

Use of Large Language Models (LLMs) in Qualitative Analysis: Evaluating LLMs as Assistive  
Coding Agents

by

Mikayla Brooke Neeb

A thesis  
presented to the University of Waterloo  
in fulfilment of the  
thesis requirement for the degree of  
Master of Science  
in  
Public Health Sciences

Waterloo, Ontario, Canada, 2026

©Mikayla Brooke Neeb 2026

## Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Statement of Contributions

Mikayla Brooke Neeb is the sole author for chapters 1, 3-5, which were written under the supervision of Dr Helen Chen. Exceptions to sole authorship are as follows:

Edson Kenzo Takei supported the development of the initial Python code, as stated in the acknowledgements below, specifically in filtering and data processing steps. Section 2.5.1 was contributed by Edson Kenzo Takei and incorporated into Chapter 2 by Mikayla Neeb.

Andrew Yang, Edson Kenzo Takei and Bing Hu developed the initial coding prompt, which helped shape the direction of the study and future versions of the code generation.

There are also those who contributed to the human evaluation phase of this study, by participating in the evaluation: Genevieve Lee, Raushan Baisalova, Edson Kenzo Takei, Sierra Radovini, Sultan Abdulkarim, Ziting Shen, and Jonathan Young-Man Lee. In addition, Sierra Radovini, Sultan Abdulkarim, Shaheena Fakir and Edson Kenzo Takei assisted in re-coding the primary dataset.

## Abstract

**Introduction:** Large language models (LLMs) are increasingly used to support qualitative research, yet robust methods to evaluate the quality of LLM-generated codes remain underdeveloped. Existing approaches often rely on comparisons to human ground truth or custom evaluative methods, limiting cross-study comparisons. This study examines whether LLMs can function as assistive qualitative coding agents and introduces the Consistency Relevance Distinction Specificity (CReDS) framework as a structured approach to evaluating LLM-generated codes without the need for a comparative codebase.

**Methods:** Two social media datasets were employed as validation sets to systematically develop and test approaches for evaluating LLM-generated inductive codes. Codes were generated using GPT-4o-mini and assessed through an iterative evaluation process. Initial assessment relied on conventional quantitative similarity metrics (e.g., cosine similarity); however, limitations in capturing qualitative distinctions prompted the incorporation of structured human evaluation. This process led to the development of the CReDS framework, comprising Consistency, Relevance, Distinction and Specificity, as a more comprehensive evaluative method. Targeted exploratory analyses further examined evaluative performance under specific conditions, further investigating the evaluative methods explored in this study.

**Results:** LLM-generated codes aligned closely with human codes across both datasets, with overall semantic match rates ranging from 74-83%. At the text level, 65-95% of inputs had at least one LLM-generated code judged appropriate by human reviewers. CReDS scores revealed strong alignment to human-generated codes, with strong overlap across all dimensions. However, LLM-generated codes showed reduced specificity, and the CReDS framework observed conservative scoring behaviour. Despite these limitations, CReDS effectively surfaced systematic strengths and weaknesses in LLM outputs.

**Conclusions:** These findings indicate that LLMs can reliably support early state qualitative coding when used as assistive tools under human oversight. The CReDS framework offers a transparent and scalable method for evaluating LLM-generated codes that align with qualitative principles while supporting iterative model development. This study contributes to a measurable and scalable platform for responsible human-AI collaboration in qualitative analysis and highlights directions for refining evaluation frameworks in future work.

## Acknowledgements

I would like to acknowledge the support and contributions of my supervisor and committee members. Your ongoing support and contributions have made this study fruitful.

I would also like to thank all those who assisted and contributed their time and skills to re-code the primary data set and complete the human evaluation.

Additionally, I would like to thank Gaya Bin Noon and Bobbi Rotolo for providing this study with data from their study *Public Attitudes Towards Vaccine Passports in Alberta During the 'Pandemic of the Unvaccinated': A Qualitative Analysis of Reddit Posts*, including comment-code pairs that were retrieved and coded by the authors. Furthermore, I would like to thank Nauman Shakeel for providing this study with data from his study *Examining the public discourse around the closures of safe injection sites: thematic analysis*, including comment-code pairs that were retrieved and coded by the authors.

## Dedication

This thesis is dedicated to my loving, supportive husband, without whom I never would have completed this work, nor would I have had the courage to start. For B, and what is to come.

## Table of Contents

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| Author's Declaration.....                                                | ii  |
| Statement of Contributions .....                                         | iii |
| Abstract .....                                                           | iv  |
| Acknowledgements.....                                                    | v   |
| Dedication .....                                                         | vi  |
| List of Figures .....                                                    | ix  |
| List of Tables .....                                                     | x   |
| List of Abbreviations .....                                              | xi  |
| 1. Background.....                                                       | 1   |
| 1.1 Introduction .....                                                   | 1   |
| 1.2 Prior Work.....                                                      | 4   |
| 1.3 Study Objectives .....                                               | 9   |
| 2. Methods.....                                                          | 10  |
| 2.1 Development of the CReDS Framework .....                             | 10  |
| 2.2 Data Source and Data Extraction .....                                | 11  |
| 2.3 Experiment Pipelines.....                                            | 12  |
| 2.3.1 Test Case One: Alberta Vaccine Passport Discourse .....            | 12  |
| 2.3.2 Test Case Two: Ontario Safe Injection Site Closure Discourse ..... | 14  |
| 2.4 Generating LLM Codes.....                                            | 16  |
| 2.4.1 Iterative Temperature Test.....                                    | 21  |
| 2.4.2 Seed Stability Test.....                                           | 21  |
| 2.5 Data Filtering.....                                                  | 21  |
| 2.5.1 Test Case 1: Alberta Vaccine Passport Discourse.....               | 21  |
| 2.5.2 Test Case 2: Ontario Safe Injection Site Closure Discourse .....   | 22  |
| 2.6 Evaluation Techniques .....                                          | 22  |
| 2.6.1 Initial Quantitative Evaluation of the LLM-generated Codes .....   | 22  |
| 2.6.2 Human Evaluation of LLM-generated Codes .....                      | 23  |
| 2.6.2.1 Test Case 1: Alberta Vaccine Passport Discourse .....            | 23  |
| 2.6.2.2 Test Case 2: Ontario Safe Injection Site Closure Discourse.....  | 24  |
| 2.6.3 Exploratory Investigations into Data.....                          | 24  |
| 2.6.3.1 Duplicated Coding Investigation .....                            | 24  |
| 2.6.3.2 Sarcasm Investigation .....                                      | 24  |
| 2.6.3.3 Specificity in Extreme Performers Analysis.....                  | 25  |

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 3. Results.....                                                               | 26 |
| 3.1 CReDS Framework .....                                                     | 26 |
| 3.2 Test Case 1: Alberta Vaccine Passport Discourse .....                     | 31 |
| 3.2.1 Generated Codes.....                                                    | 31 |
| 3.2.2 Quantitative Evaluation of LLM-generated Codes .....                    | 31 |
| 3.2.3 CReDS Evaluation.....                                                   | 33 |
| 3.2.4 Human Evaluation of LLM-generated Codes .....                           | 34 |
| 3.3 Test Case 2: Ontario Safe Injection Site Closure Discourse.....           | 36 |
| 3.3.1 Generated Codes.....                                                    | 36 |
| 3.3.2 Iterative Temperature Test.....                                         | 36 |
| 3.3.3 Seed Stability Test.....                                                | 37 |
| 3.3.4 CReDS Evaluation of LLM-generated Codes .....                           | 38 |
| 3.3.5 Human Evaluation of LLM-generated Codes .....                           | 39 |
| 3.4 Probing Analyses.....                                                     | 40 |
| 3.4.1 Duplicated Coding Analysis.....                                         | 40 |
| 3.4.2 Sarcasm Analysis.....                                                   | 43 |
| 3.4.3 Specificity Analysis.....                                               | 47 |
| 4. Discussion.....                                                            | 50 |
| 4.1 LLMs as Agents Compared to Humans .....                                   | 50 |
| 4.2 Implementation, Performance and Implications of the CReDS Framework ..... | 51 |
| 4.3 Ethical Implementation .....                                              | 52 |
| 4.4 Positionality and Reflexivity.....                                        | 52 |
| 4.5 Strengths.....                                                            | 53 |
| 4.6 Limitations .....                                                         | 54 |
| 4.7 Impacts and Recommendations.....                                          | 55 |
| 4.8 Future Work .....                                                         | 56 |
| 5. Conclusions.....                                                           | 57 |
| References.....                                                               | 59 |
| Appendices.....                                                               | 65 |
| Appendix 1. Supplementary Tables and Figures .....                            | 65 |
| Appendix 2. Supplementary Background Information .....                        | 69 |

## List of Figures

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Figure 1 Study Process-Map, Test Case One.....                                      | 13 |
| Figure 2 Test Case Two Study Process Map .....                                      | 15 |
| Figure 3 Heatmap of the CSA score equation .....                                    | 28 |
| Figure 4 Distribution of CReDS Dimension Scores by Code Source, Test Case One ..... | 34 |
| Figure 5 Distribution of Dimension Scores per Temperature.....                      | 37 |
| Figure 6 Distribution of Dimension Scores per Seed Run.....                         | 38 |
| Figure 7 Distribution of CReDS Dimension Scores by Code Source, Test Case Two ..... | 39 |
| Figure 8 CReDS Dimension Scores by Sarcasm Label, LLM.....                          | 45 |
| Figure 9 CReDS Scores by Sarcasm Label for Human-Generated Codes .....              | 46 |

## List of Tables

|                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------|----|
| Table 1. Review of Relevant Literature .....                                                                   | 6  |
| Table 2 Evaluative Rubric for Relevance .....                                                                  | 28 |
| Table 3 Evaluative Rubric for Specificity .....                                                                | 30 |
| Table 4 Crude Generated Codes Count .....                                                                      | 31 |
| Table 5 Summary of Cosine Similarities LLM Code to Original Code and Text Results, N = 370<br>.....            | 32 |
| Table 6 Summary of Cosine Similarities, Human Benchmark, N = 370.....                                          | 32 |
| Table 7 Summary of CReDS Framework Evaluation, Test Case One N =370 .....                                      | 33 |
| Table 8 Summary of Human Evaluation Percentages per Agent, N = 37 x 3 .....                                    | 35 |
| Table 9 Example of Non-Matching Codes .....                                                                    | 35 |
| Table 10 Counts of LLM and Human-generated Codes .....                                                         | 36 |
| Table 11 Summary of CReDS Framework Evaluation, Test Case Two.....                                             | 38 |
| Table 12 Summary of Human Evaluation Percentages .....                                                         | 40 |
| Table 13 Summary of Duplicate Analysis, N = 28 .....                                                           | 41 |
| Table 14 Example of Duplicate Codes for Human and LLM-generated Codes.....                                     | 42 |
| Table 15 Summary of Sarcasm Analysis .....                                                                     | 43 |
| Table 16 Reasons for Code Inappropriateness, N=52 .....                                                        | 43 |
| Table 17 Examples of Sarcastic Input Text and Corresponding Codes.....                                         | 44 |
| Table 18 Percent Differences Between Sarcastic and Non-Sarcastic CReDS Scores Among<br>Coder Groups .....      | 46 |
| Table 19 Dimension Specific Differences Between Sarcastic and Non-Sarcastic CReDS Scores,<br>LLM - Human ..... | 47 |
| Table 20 Summary of Specificity Analysis, N = 60 .....                                                         | 48 |
| Table 21 Top and Bottom Scoring Codes within Specificity Dimension.....                                        | 48 |
| Table 22 Difference in Scores in Top and Bottom Codes by Evaluator, N = 60.....                                | 49 |

## List of Abbreviations

CAQDAS Computer-assisted qualitative data analysis software

CReDS Consistency Relevance Distinction Specificity Framework

GPT Generative Pre-trained Transformer

LLM Large Language Models

IPA Interpretive Phenomenological Analysis

# 1. Background

## 1.1 Introduction

Qualitative research aims to examine various phenomena by seeking information regarding individuals' experiences and perspectives through the systematic collection and analysis of non-numeric data (Oranga & Matere, 2023). This data may be collected through interviews, observations, historical data obtained from databases or registries, or other textual or visual sources. Thematic analysis, a foundational method of qualitative research, investigates and reports patterns, commonly referred to as themes, within qualitative data (Braun & Clarke, 2006). These themes are summations of initial codes, which are labels or short representations capturing salient features of the data investigated. Thematic analysis is a widely used qualitative data analysis method among qualitative researchers due to its flexibility and accessibility. Thematic analysis differs from other analytic methods that seek to describe patterns across qualitative data, such as, but not limited to, thematic decomposition analysis, interpretive phenomenological analysis (IPA) and grounded theory. Both IPA and grounded theory are theoretically bound and grounded in data, whereas thematic analysis is not connected to any pre-existing theoretical framework (Braun & Clarke, 2006). Common practices include open or initial coding, where researchers identify and name essential concepts and patterns. Then, researchers compare data and codes until they eventually condense code into categories, categories into themes and themes into findings (Bingham, 2023). Braun and Clarke's framework, while widely used, does not apply to each thematic analysis type equally. However, in practice, thematic analysis typically begins with a thorough reading of the data, during which the researcher becomes immersed in the content and identifies meaningful units of text relevant to the research questions (Braun & Clarke, 2006). These units, often sentences or phrases, are labelled with concise descriptors, a process referred to as coding (Saldaña, 2013). This initial stage may involve line-by-line or segment-by-segment coding and is followed by a recursive process of reviewing, comparing and refining codes to ensure they accurately capture the nuances of the data (Babchuck, 2019; Bradley et al., 2007). As patterns begin to emerge, similar codes are clustered into broader categories, which are then refined and elevated into overarching themes that capture the essence of the dataset. These themes are not simply summaries of content but represent interpretive patterns that respond to the analytic goals of the study (Nowell et al., 2017). While the specific phases may vary slightly depending on the analytic tradition followed, these core steps, familiarisation, coding, categorising and theme development, form the foundation of most thematic analyses.

Themes or patterns in data can be identified in one of two primary ways in thematic analysis: inductively or deductively. Deductive, or theoretical thematic analysis, is explicitly analytically driven and can be considered a "top-down" approach. This type of analysis can be used to organise data into predetermined categories from literature or theory. Researchers will begin with a specific research question and pre-established codes derived from the literature or theory, and apply them to the data (Bingham, 2023). Alternatively, inductive analysis focuses on generating themes from the data without relying on a predefined theoretical model, a process

sometimes referred to as a “bottom-up” approach (Braun & Clarke, 2006). Codes and categories are not predetermined; rather, they are identified and named as researchers read the data (Bingham, 2023). Whatever approach is used, the process of coding relies heavily on researchers’ interpretation, and therefore can potentially benefit from strategies that strengthen rigour.

Qualitative research is characterised by its depth, interpretive complexity and attention to meaning-making within specific contexts. However, these strengths create persistent methodological challenges (Ahmed et al., 2025). The entire process of coding is labour-intensive and often constrained by personnel, time and available funding (Saldaña, 2013; Han et al., 2025). These challenges limit the scalability of qualitative inquiry, often requiring teams of researchers, who in turn require extensive coordination and training to establish shared understandings (McAlearney et al., 2023). Although not universally applicable, the involvement of multiple coders can enhance the consistency and reliability of the generated codes and themes (MacQueen et al., 1998). In practice, maintaining rigour across coders demands ongoing dialogue, reflexivity and iterative review, activities that, while vital for ensuring analytical depth, also amplify the resources required (Barbour, 2001). These demands underscore the broader challenge facing the field: balancing interpretive richness and the need for greater efficiency and scalability. Considering this, it is important to determine how to do qualitative research not only faster, but also more efficiently and effectively.

Computational tools, including computer-assisted qualitative data analysis software (CAQDAS), have long sought to address aspects of these problems by facilitating data organisation and retrieval. Prior to the 1960s, qualitative researchers relied on manual pen-and-paper methods to analyse their data (Wolski, 2018). When early mainframe computers were introduced in the 1960s, researchers began using them for qualitative content analysis. While some rudimentary packages were starting to be developed in the 1970s, researchers in the 1980s began developing software for the analysis of qualitative data (Wolski, 2018). Despite initial concerns and skepticism, such programs were developed and commercialised in the 1990s, and qualitative research aids, such as NVivo™, have since become popular within the industry (Wolski, 2018). Nvivo™, with roots in one of the first CAQDAS, is designed to support teams and can accommodate a wide range of qualitative research designs. Despite these technological improvements, many researchers report challenges using the software (Zamawe, 2015), and such tools still rely on manual coding and human interpretation. This creates an opportunity for advancements that improve the user experience, such as new technology-assisted qualitative coding.

To understand how LLMs may function as assistive tools in qualitative analysis, it is necessary to briefly outline their underlying technical properties and modes of interaction. LLMs are a category of foundation models based on the transformer architecture that excel at handling sequential data such as text input (Vaswani et al., 2017). These models consist of multiple layers of neural networks, each with parameters that can be fine-tuned and further enhanced by an

attention mechanism, which helps the model focus on the relevant parts of the input data (Vaswani et al., 2017). These techniques offer advances over earlier natural language processing methods, which often relied on rule-based systems or statistical methods that required manually customised features (Stryker & Holdsworth, 2024). One of the key features of LLMs is their adaptability and flexibility through prompting and parameter tuning. These models are not retrained for each task; instead, they are used through prompting, a technique in which instructions are crafted to guide the model's responses (Sahoo et al., 2024). Prompting enables flexible adaptation to diverse research needs, and this adaptability can be further shaped using configurable parameters. Most commonly, the temperature parameter can be adjusted to control the randomness of model outputs. Ranging from 0.0 to 1.0, the temperature parameter allows for highly specific configurations. For lower values (e.g., 0.1), a model produces more deterministic responses, whereas higher values reduce the likelihood of a deterministic response and thereby increase creativity and variability (Peeperkorn et al., 2024). This makes temperature an important parameter for aligning model behaviour in research; low temperatures are often preferred for tasks requiring consistency and precision, whereas higher values may be useful for exploratory analysis. This allows researchers to tailor LLM to explore different model behaviours and fine-tune outputs for different analytic tasks.

The emergence of LLMs introduces a new methodological frontier. By using LLMs, researchers can leverage AI models to streamline thematic analysis (Dunivin, 2024; Randerson et al., 2025; Zhang et al., 2024; Raza et al., 2025). LLMs are extensively trained on a vast variety of data, making them capable of reading and generating natural language (IBM, 2023). They have the ability to generate coherent and contextually appropriate responses, summarise text, translate languages and answer questions for users; all by identifying intricate patterns in human language (IBM, 2023). This allows LLMs to avoid the limitations human coding teams face, which are often restricted to smaller sample sizes (Dunivin, 2024). As such, many studies have shown that LLMs can be utilised as independent agents for thematic analysis on a consistent basis by adjusting model parameters (Dunivin, 2024; De Paoli, 2024).

Since temperature and other parameters are easily adjusted through user-friendly interfaces, even non-specialists can now tailor model outputs to suit specific needs. With the public availability of LLMs such as OpenAI's ChatGPT, their use has expanded dramatically in both academic and non-academic settings, offering new opportunities for methodological innovation. However, the integration of LLMs into qualitative research is still in early development, and concerns persist among researchers. A range of concerns have been expressed regarding the integration of LLMs into research, particularly qualitative research, as the rapid development of these technologies has outpaced the development of clear ethical guidelines and best practices (Schroeder et al., 2025). Key issues include model bias and interpretability, challenges in validating and evaluating model outputs, implications for participant privacy, and the risk of unequal access or adoption across institutions (Schroeder et al., 2025). A perceived challenge among researchers is that the use of LLMs may lead to a lack of deep, nuanced engagement with data and participants that traditional qualitative research emphasises (Schroeder

et al., 2025; Ornelas et al., 2025). Despite these concerns, many researchers have begun using LLMs in their work (Schroeder et al., 2025). In interviews, researchers noted that LLMs could be helpful for tasks such as generating recruitment materials, supporting non-native English speakers with writing and proofreading and, most notably for this study, generating initial codes (Schroeder et al., 2025).

The integration of computational systems into qualitative inquiry introduces a series of epistemological tensions. LLMs, grounded in probabilistic reasoning, produce text by identifying and extending linguistic patterns rather than engaging in true interpretive understanding. This mode of operation stands in contrast to the interpretivist traditions of qualitative research, in which meaning is constructed through context, reflexivity, and researcher engagement. As computational methods gain traction in the social and health sciences, questions emerge about whether these systems can contribute meaningfully to interpretive practice or whether their use offers greater risks (Schroeder, 2025). These tensions are not merely technical but also methodological, challenging qualitative researchers to reconsider what constitutes rigorous interpretation and collaboration in an era of human-machine analysis (Roberts et al., 2024; Reeping et al., 2025).

Qualitative research is increasingly conducted within data-rich environments that demand both interpretive sensitivity and methodological scalability. As digital and social data become more prevalent, the field faces growing tension between the depth of traditional qualitative methods and the scale of available data (Dunivin, 2025; Hunt et al., 2011). Scholars such as Braun and Clarke (2006) have emphasised that qualitative analysis must remain grounded in meaning-making, context, and reflexivity rather than in mechanical procedures. Yet, the expansion of computational tools, including text mining and natural language processing, has prompted new questions about how automation can intersect with interpretive inquiry. This study situated itself within this evolving landscape, exploring how LLMs might serve not as replacements for interpretive labour but as extensions of it.

This study uses data from previous qualitative studies: *Public Attitudes Towards Vaccine Passports in Alberta During the “Pandemic of the Unvaccinated”* (Rotolo et al., 2023) and *Examining the public discourse around the closures of safe injection sites: thematic analysis* (Shakeel et al., 2025). These two datasets are used as a validation case for the proposed evaluation technique, where each dataset and its surrounding social context provide valuable opportunities to explore and assess the utility of the proposed evaluation in a real-world setting.

## 1.2 Prior Work

Table 1 summarises recent studies that explore the use of LLMs for qualitative coding. Each study is characterised by the tools and models used, data sources analysed and evaluation techniques used. Together, these studies illustrate the evolving capabilities and methodology of LLMs in supporting or automating thematic analysis across various research contexts. To contextualise this study, a literature search was conducted across multiple databases. Relevant

publications were identified using keyword combinations such as “qualitative analysis”, “thematic analysis” or “qualitative coding” in conjunction with “LLM\*” or “large language model\*”. Searches were primarily performed on arXiv to capture emerging work in machine learning and computational methods, and on PubMed to identify peer-reviewed studies with applications in health and social sciences. The search included papers published from 2022 to 2025. Works were included if they directly examined the use of LLMs for qualitative coding or thematic analysis, either as a primary method or part of a mixed-methods approach. Studies were excluded if they were master’s or doctoral theses, undergraduate papers, position papers, conceptual frameworks without empirical testing, or unreviewed preprints.

These studies collectively explored the use of LLMs for thematic qualitative coding, demonstrating varied levels of success across frameworks, analytic goals and data types. Despite their different approaches in methodological design, model selection and evaluation criteria, the studies converge on several key insights. LLMs can reliably support inductive coding, especially with structured, iterative and collaborative conditions. These studies showed that LLMs can approach or even surpass human performance in inductive coding, as in Montes et al. (2025), where 61% of evaluators preferred LLM-generated codes. Despite these successes, thematic analysis utilising LLMs still requires careful prompt design and human judgment moving forward. Limitations remain in terms of code specificity compared to human-generated codes, with LLM-generated codes tending to be more generalised and superficial (Raza et al., 2025). Challenges related to hallucinations, code redundancy and replicability are also pressing concerns that have been seen within LLM-related research since its emergence (Deng et al., 2023).

A notable gap in the literature is the absence of standardised evaluation metrics for comparing LLM and human coding performance beyond basic measures such as inter-coder reliability or comparisons with cosine similarity. While some studies employ metrics such as Cohen’s kappa or human alignment percentages, many rely on custom qualitative assessments, limiting cross-study comparability (Chen et al., 2024; Oksanen et al., 2025). These evaluations rely heavily on human baseline comparisons, which may not pose an undue challenge for studies with an existing human comparative codebase but do so for exploratory studies without such data. An efficient, independent evaluation of LLM-generated codes could streamline iterative LLM development in thematic analysis for customised, effective assistive agents. These gaps highlight the need for structured frameworks to evaluate not only the quality of the generated codes but also the process by which they are produced. Beyond the absence of standardised metrics, the literature also lacks evaluative frameworks that are explicitly grounded in qualitative research traditions and epistemological assumptions. Many existing evaluation approaches borrow from quantitative validation, emphasising agreement, similarity, or convergence rather than criteria such as analytical usefulness or reflexive transparency that are central to qualitative inquiry. This misalignment risks evaluating LLM-generated codes using criteria that are not

Table 1. Review of Relevant Literature

| Study                              | Tools                                                                                           | Data Source                                                                                    | Evaluation Method                                                                                 |
|------------------------------------|-------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Katz, Fleming and Main (2022)      | GATOS workflow; Llama-3.1-70B, WizardLM2-8x22B, Mitral-Nemo-12B, Llama-3.1-8B, Mistral-22B-2409 | Three simulated datasets                                                                       | Comparison against human-generated codes; cosine similarity                                       |
| Dai et al. (2023)                  | GPT-3.5; human-in-the-loop interactive framework                                                | Open datasets (music shuffling on personal devices, password manager use)                      | Comparison against human-generated codes; cosine similarity                                       |
| Drápal, Westermann, Savelka (2023) | GPT-4, tailored prompt pipeline                                                                 | 785 descriptions from criminal court records                                                   | Comparison against human-generated codes; manual inspection                                       |
| Dunivin (2024)                     | GPT-3.5, GPT-4, CoT and zero-shot prompting                                                     | New York Times article dataset                                                                 | Comparison against human-generated codes; Cohen’s kappa (interrater reliability)                  |
| Zhang et al. (2024)                | QualiGPT                                                                                        | Simulated and real datasets: focus group on transitioning to remote work, social media dataset | Comparison against human-generated codes; Cohen’s kappa and Fleiss’ kappa                         |
| Mathis et al. (2024)               | Whisper (to convert audio to text), LLaMA-2-70B-Instruct, Sentence-T5-xxl                       | Semi-structured interviews in a psychiatric setting                                            | Comparison against human-generated codes; Jaccard coefficient                                     |
| De Paoli (2024)                    | GPT-3.5 Turbo-16K                                                                               | Interview transcripts                                                                          | Comparison against human-generated themes; cosine similarity and LLM prompt                       |
| Chen et al. (2024)                 | Gecko-768, GPT-3.5, GPT-4o-0513, LLaMA3-70B,                                                    | Two human-computer interaction datasets                                                        | Aggregated comparison against human-generated codes; cosine distance, hierarchical clustering and |

|                                    |                                                                                               |                                                              |                                                                                                                            |
|------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|                                    | Mixtral 8x22B, Claude models (Haiku, Sonnet)                                                  |                                                              | calculated metrics (coverage, density, novelty and divergence)                                                             |
| Gao et al. (2025)                  | MindCoder (web app with CoT prompting)                                                        | Six varied datasets                                          | Comparison against human-generated codes; Jaccard similarity and manual inspection                                         |
| Raza et al. (2025)                 | GPT-4o mini, LangChain, LLT-TA pipeline                                                       | Nine focus group transcripts                                 | Comparison against human-generated codes: manual inspection, pairwise cosine similarity, Jaccard similarity and Hit Rate   |
| Randerson (2025)                   | GPT-4                                                                                         | 108 media items (alcohol policy discourse network)           | Comparison against human-generated codes; manual inspection                                                                |
| Oksanen, Lucero, Hämäläinen (2025) | LLMCode, qualitative agreement metrics (Intersection over union, Modified Hausdorff distance) | Two studies investigating game design research               | Comparison against human-generated codes; intersection over union and Jaccard Index                                        |
| Parkington et al (2025)            | GPT-4o                                                                                        | 99 semi-structured interview transcripts                     | Comparison against human-generated codes; resources used, thematic saturation, analytic quality (reliability and validity) |
| Nyaaba et al. (2025)               | GPT-4 Turbo via API with a prompt-based Python script                                         | Semi-structured interview data                               | Comparison against human-generated codes; manual inspection                                                                |
| Hila and Hauser (2025)             | ChatGPT, RoBERTa and bidirectional LSTM                                                       | US Supreme Court case summaries into 21 major policy domains | Standard classification metrics; accuracy, F1, Cohen's kappa, Krippendorff's alpha                                         |

|                     |                                                   |                                                                              |                                                                                                                                                                                                                                                                          |
|---------------------|---------------------------------------------------|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Zhao and Liu (2025) | Claude, GPT 4o, DeepSeek, Gemini, llama4-maverick | 71 interview segments across 10 categories                                   | LLM self-confidence, comparison against human-generated codes; Cohen's kappa                                                                                                                                                                                             |
| Borse et al. (2025) | ChatGPT-4o, ChatGPT-4.5-preview                   | Audio transcripts                                                            | Comparison against human-generated codes; Cohen's kappa                                                                                                                                                                                                                  |
| Yi et al (2025)     | GPT-4o Supervised fine-tuned, multi-agent systems | Deidentified interview transcripts from 9 focus groups                       | Comparison against human-generated codes by “traditional metrics” (lexical overlap, BLEU, METEOR, ROGUE-L, fuzzy match and cosine similarity) and thematic analysis specific metrics (credibility, confirmability, dependability and transferability), manual inspection |
| Sharma et al (2025) | DeTAILS toolkit, Gemini-2.5-pro-preview-03-25     | Reddit social media data                                                     | Manual inspection, comparison to human baseline                                                                                                                                                                                                                          |
| Montes et al (2025) | Claude sonnet 4, ChatGPT-o4, ChatGPT-o3 mini      | 15 semi-structured interviews exploring the well-being of software engineers | Comparison against human-generated codes; manual inspection using a rubric based on Braun and Clarke                                                                                                                                                                     |

well-suited to inductive forms of thematic analysis. Introducing an evaluative framework that is informed by qualitative theory enables assessment to move beyond surface-level alignment and toward consideration of how codes function analytically and how model behaviour interacts with interpretive decision-making. Such frameworks offer a structured yet flexible means of evaluating LLM-assisted coding in a manner that is both methodologically rigorous and epistemologically coherent, thereby addressing a critical gap in current research.

### 1.3 Study Objectives

This project has two primary objectives: first, to explore the use of LLMs as assistive agents in qualitative coding, including both technical and interpretive capacities, and second, to critically evaluate the quality of codes generated by LLMs. Through this evaluation, the feasibility of integrating LLMs into such research methodologies will be assessed. This study does not seek to replace the human analytic process but to investigate how LLMs may be positioned within it and to identify the challenges that arise from their use. In doing so, this study contributes to ongoing debates about how rigour, reflexivity and collaboration might be redefined in human-AI collaborative qualitative research. In pursuit of these goals, this study poses the following questions:

1. In what ways can qualitative methodologies inform the evaluation and refinement of LLM-generated qualitative codes?
2. What measures can be used to evaluate the quality of LLM-generated qualitative codes without comparison to researcher-generated codes for the same input

## 2. Methods

### 2.1 Development of the CReDS Framework

The CReDS Framework was developed to address the lack of qualitative theory-grounded approaches for evaluating LLM-generated codes. Its development was informed by a synthesis of expert consultation and an extensive review of relevant literature. Specifically, the framework emerged through iterative engagement with an established expert qualitative researcher, whose methodological expertise informed both the conceptual grounding of the framework and the identification of key evaluative dimensions relevant to qualitative coding.

The literature informing the development of CReDS was drawn from two primary sources. First, a curated set of foundational qualitative methodology texts and articles was provided by a qualitative research expert, reflecting established traditions in thematic analysis, reflexivity and qualitative rigour. Second, this initial corpus was supplemented through systematic literature searches conducted by the author. These searches included reference chaining from relevant publications, review of papers cited within methodological and applied qualitative studies and targeted searches of qualitative research journals indexed in PubMed. In parallel, literature addressing the use of LLMs in qualitative and mixed-methods research was identified, primarily through PubMed and arXiv, with particular attention to studies proposing or employing evaluation strategies for LLM-assisted analysis. The sources specifically related to LLMs included in this review are summarised in Table 1.

The dimensions of the CReDS framework were guided by a central question: *what constitutes a “good code in qualitative research?”* Within qualitative traditions, there are no fixed or universally accepted criteria for evaluative code quality, as coding practices are inherently shaped by epistemological stance, analytical goals and interpretive context (Cena et al., 2024). Rather than seeking strict or binary definitions of correctness, qualitative methodology emphasises transparency, analytical usefulness and coherence within a given interpretive framework (Braun & Clarke, 2006).

In developing the CReDS dimensions, this absence of rigid guidelines necessitated the interpretation of general coding principles and recurring evaluative concerns articulated across qualitative methods literature. These guidelines and concerns informed the selection and definition of the core dimensions of the CReDS framework. Through this process, CReDS was designed to provide a structured yet flexible framework for evaluating LLM-generated codes in a manner that aligns with qualitative research principles while remaining applicable across varied analytical contexts.

Following the identification of the core CReDS dimensions, the evaluation criteria and scoring mechanisms for each dimension were developed through an iterative and exploratory process. Initial efforts focused on identifying existing evaluation metrics from literature that could be adapted for use in this context. These included commonly reported quantitative

measures such as ROUGE, F1 score and accuracy, as well as other similarity and alignment-based metrics used in prior studies of LLM performance. However, these metrics were found to align poorly with the interpretive aims and assumptions of qualitative coding, as they primarily emphasise lexical overlap or convergence rather than analytical meaning or conceptual contribution.

In parallel, candidate evaluation approaches were informed by discussions with supervisory committee members and laboratory colleagues, which provided additional perspectives on potential evaluative strategies and highlighted practical considerations related to implementation. Despite these inputs, few existing metrics adequately translated the CReDS dimensions into a feasible Python-based evaluation pipeline. As a result, custom evaluation approaches were developed for each dimension, as fully described in section 3.1. Among these custom evaluation approaches, LLM-evaluated structured rubrics were employed. These rubrics were designed to operationalise qualitative evaluative concepts in a transparent and replicable manner, while allowing for an automative approach. The definitions and scoring criteria for each rubric were grounded in established qualitative methodology literature and, where relevant, in linguistic definitions of the underlying constructs. See Section 3.1 for the complete rubrics.

## 2.2 Data Source and Data Extraction

There are two data sources for this study, both of which are Reddit comments from Canadian-focused subreddits. These datasets were provided to this study for use from two previous studies. Both studies provided insight into public perceptions during controversial policy changes within Canada, such as vaccine passports and safe injection sites. The controversial nature of the topic, combined with the challenging input for code generation, enables a thorough evaluation of the capabilities and limitations of LLM qualitative coding.

The first data source for this project is Reddit comments from the subreddit *r/Alberta* regarding perceptions of mandatory vaccine passports implemented in September 2021. This data was previously gathered using the Python Reddit API Wrapper (PRAW) (Boe, 2014), traditionally coded by humans and stored in NVivo™ file format (Rotolo et al., 2023). A portion of the original dataset, including both the original comment and corresponding codes, was retained by the authors and provided to this study for use. All independently exported codes were extracted from each PDF file and merged into a single original human-coded Excel file, which could be used to generate the LLM initial codes. This resulted in a total of 414 extracted text-original code pairs, further referred to as data points, before any data processing to remove duplicates and errors in formatting. After data processing, 370 data points remained.

The second data source for this project is Reddit comments from the subreddit *r/Ontario* regarding public perceptions of supervised injection site closures between August and December 2024. This data was gathered using Python's Pushshift API (2019), traditionally coded by humans, stored in Excel format and provided to this study for use (Shakeel et al., 2025). This resulted in a total of 645 original extracted thread title-comment-code pairs, further referred to as

data points, before any data processing. At the human evaluation phase, 550 data points remained for sampling due to the reduced size of the corresponding human-generated codebase.

## 2.3 Experiment Pipelines

### 2.3.1 Test Case One: Alberta Vaccine Passport Discourse

Test Case One represents the initial exploratory phase of this study and was designed to establish and evaluate the analytical pipeline that is further developed through this study. This phase focused primarily on quantitative evaluation, with particular attention to the influence of model temperature on code generation. By varying temperature settings and examining their effects on coding outputs, Test Case One provided foundational insights that informed the methodological decisions and analytical approaches applied in subsequent stages of this study.

Figure 1 illustrates the overall process flow for Test Case One, including paraphrased file names and the actions taken at each step.

First, the original raw data is obtained and converted to a usable .csv format. This data is then used to generate initial codes using three agents operating at three distinct temperatures (0.1, 0.5, 1). The outputs are subsequently filtered and processed by a quantitative evaluation using cosine similarity calculations. A random 10% sample is subsequently selected for the human evaluation. The human evaluation, described in full in Section 2.4.3, produces qualitative evaluation data, that are combined with the quantitative evaluation for the full analysis of Test Case One.

While not part of the original workflow for Test Case One, an automated evaluation was conducted retrospectively using the CReDS framework. Both human-generated and LLM-generated codes were assessed using CReDS to enable a structured comparison of coding quality and characteristics. This post hoc evaluation was undertaken to support comparability with Test Case Two, in which the CReDS-based evaluation was incorporated directly into the analytical pipeline, and to provide additional quantitative context for interpreting differences across test cases.

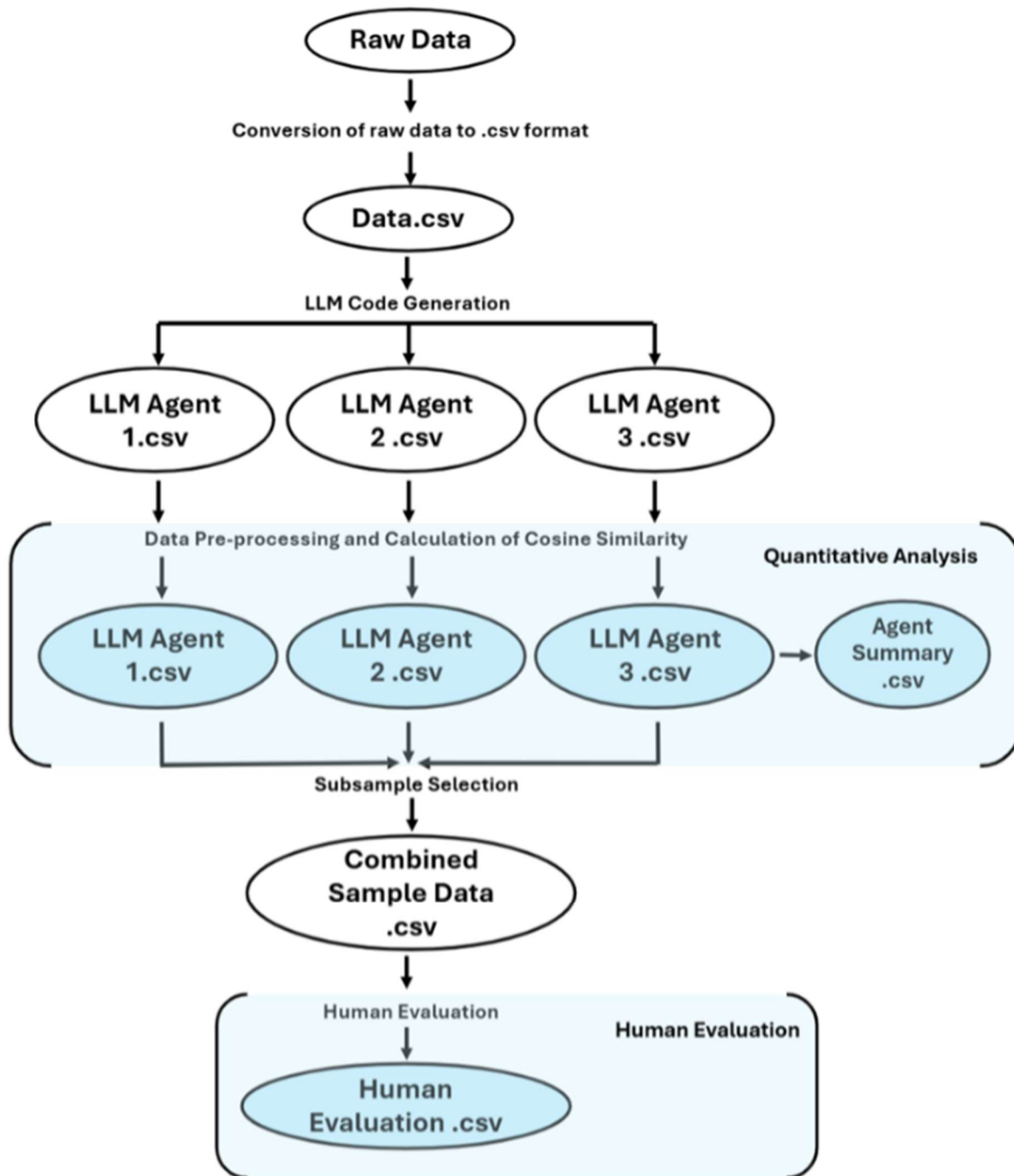


Figure 1 Study Process-Map, Test Case One.

This figure shows the paraphrased file names and the process flow for Test Case One. This process takes the raw data, generates initial codes for three agents with three temperatures (0.1, 0.5, 1) and quantitatively processes these outputs. Then, a random 10% sample is selected to proceed to the human evaluation. Quantitatively processed data and human evaluation results are combined for the final evaluation.

### 2.3.2 Test Case Two: Ontario Safe Injection Site Closure Discourse

Test Case Two builds directly on the findings and quantitative limitations identified in Test Case One and represents a developmental phase of the study. This phase was designed to refine the analytical pipeline through targeted prompt development in response to the specific challenges posed by a new dataset. In contrast to the primarily exploratory focus of Test Case One, Test Case Two emphasises methodological robustness, including systematic evaluation of temperature consistency and seed stability. Together, these refinements enable a more rigorous assessment of model behaviour and coding reliability under varied experimental conditions.

Figure 2 presents the updated study process used for Test Case Two, including paraphrased document names and the actions performed at each stage. The process was executed separately for the iterative temperature test and the seed stability experiment to assess consistency with respect to both the temperature parameter and seed configuration. While the overall structure follows a similar workflow to Test Case One, Test Case Two removes the separation into distinct agents and introduces an automated evaluation step. The process begins with the raw data being coded by a designated coder LLM, as in Test Case One, and then evaluated using the automated CReDS evaluation pipeline. These first two steps, coding and evaluating, are repeated for the respective iterative temperature testing and seed stability testing. Finally, data is sampled for human evaluation (30%) by stratified sampling according to the word count in the input text. All quantitative and qualitative outputs are then aggregated and summarised for final analysis.

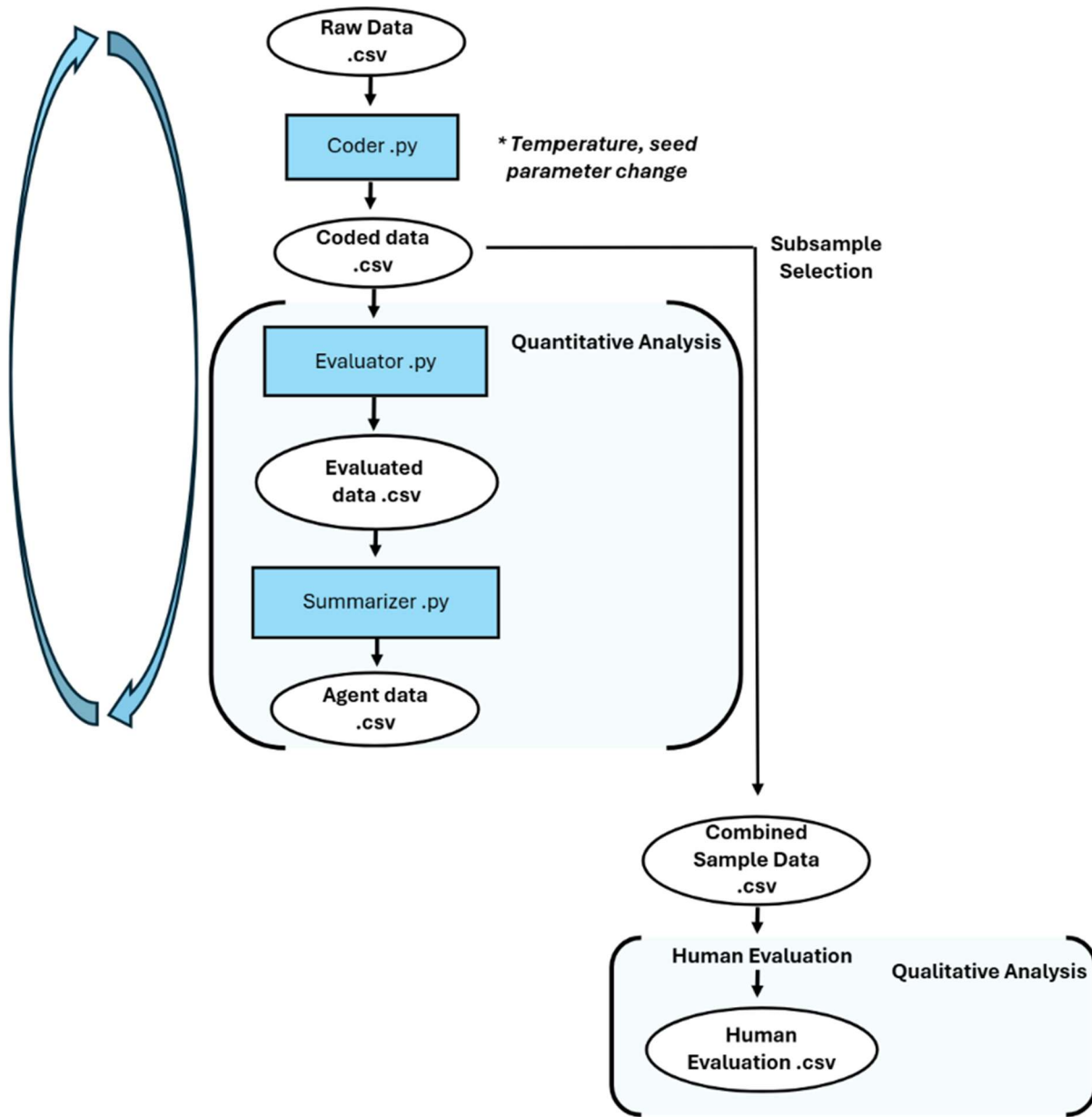


*Figure 2 Test Case Two Study Process Map*

*This diagram shows the updated study process map used for Test Case Two, including the paraphrased document names and actions taken for each step of the process. This process was executed separately for the iterative temperature test and seed stability experiment to assess consistency for the temperature parameter and the seed configuration. This process follows similar steps to Test Case One, save for the separation into separate agents. The raw data is coded by the coder LLM, evaluated by evaluating LLM and sampled for the human evaluation phase (30%) before all results are summarised for analysis.*

## 2.4 Generating LLM Codes

This project relied on step two of Braun and Clarke’s foundational methods for qualitative analysis, by generating initial codes. A summary of each of the six steps is available in Appendix 2. Figures 1 and 2 summarise the methodological processes used in this study, using paraphrased document names to illustrate the input and output of each step.

Using OpenAI’s GPT 4o-mini, initial codes were generated with an engineered prompt designed to generate all relevant codes to a piece of text, hereafter referred to as input text. The GPT-4o-mini base model was trained on data current as of October 2023 (OpenAI, 2024). This model outpaces previous versions at a fraction of the cost, showing itself to offer competitive performance. Since this work has commenced, several updated models have been released, including GPT-4.1 in April 2025 and GPT-5 in August 2025. While reported model performance metrics showed a modest improvement across updates (OpenAI, 2025), all code generation remained with GPT-4o-mini. This decision was made to maintain methodological consistency and comparison between experiments. This aligns with the study’s objectives, which focus not on maximising model performance but on evaluating the characteristics and limitations of LLM-generated codes at a baseline level.

An open-ended prompt was developed to provide the LLM with context and task-specific instructions for generating initial codes. The prompt begins by detailing the inductive qualitative coding methodology based on Braun and Clark’s (2006) framework and specifies that the LLM will be assigned only to step two: generating initial codes. The prompt also includes background framing, indicating the timeframe from which the data were scraped, the research goals and the topic of discussion. For Test Case One, the material is drawn from dates between September 1st and 30th, 2021, regarding “online opinions and beliefs towards mandatory vaccination policies in Alberta, Canada.” The prompt continues to give specific guidelines on coding. A complete copy of this prompt is provided below:

```
initial_prompt_template = ""
```

*You are a qualitative researcher performing thematic analysis inductively.*

*This is a process of coding the data without trying to fit it into a preexisting coding frame, or the researcher’s analytic preconceptions*

*A thematic analysis usually consists of six steps:*

- 1. Familiarizing yourself with your data*
- 2. Generating initial codes*
- 3. Searching for themes*
- 4. Reviewing themes*
- 5. Defining and naming themes*
- 6. Producing the report*

*You were assigned to do step 2 on generating the initial codes.*

*Another researcher already completed step 1 and after familiarizing themselves with the data they decided on the following research objective:*

*The goal of this study is to understand the online opinions and beliefs towards mandatory vaccination policies in Alberta, Canada, between 1st September and 30th September 2021, by analyzing Reddit posts and comments from the r/alberta subreddit.*

*This time period was chosen to ensure we captured opinions from before and immediately after the implementation of mandatory vaccines.*

*Analysing opinions of mandatory vaccines and vaccination policies offers valuable insight into public perception that can be used to understand the themes underlying these discussions.*

*The understanding of those themes could inform potential considerations by public health authorities surrounding mandatory vaccination policies.*

*In order to complete Step 2 you need to follow these guidelines:*

- Focus on identifying basic segments or elements of the data that can be meaningfully analyzed.*
- Look for both explicit (semantic) and underlying (latent) content that stands out.*
- A comment may have multiple codes. In those cases assign all the relevant codes to their relevant segments*
- Consider every possible theme or pattern during coding. Even if it seems unimportant at first, it might become relevant later.*
- Don't ignore data that contradicts the dominant patterns. These can offer valuable insights.*

*With the guidelines in mind give me the code(s) for this comment: \n\n[text]\n\"\"\"*

For Test Case Two, the version one prompt was further developed to include additional instructions in response to reflections on techniques used in traditional coding and the challenges encountered in the initial investigation of dataset number two.

Upon reflection, it was determined that some principles of reflexivity should be incorporated into the prompt. It is common knowledge that any person brings a history and perspective that may serve as a lens, as all qualitative research is highly contextual (Dodgson, 2019). Researchers may attempt to be objective, but definitive objectivity is neither attainable nor realistic (Dodgson, 2019). Reflexivity is the act of examining one's own assumptions and beliefs and thinking critically about how these factors influence the research process (Jamieson et al., 2023; Olmos-Vega et al., 2022). It is assumed that who the researcher is makes a difference in the findings of a study, and by describing the contextual intersecting relationships, it can increase the credibility of the findings and even deepen the understanding of the work (Dodgson, 2019). In one possible attempt to address reflexivity in the space of blackbox LLMs, it is possible that one way of knowing which lens the LLM will approach a task is to give it one explicitly. Therefore, a stigma lens was imposed on the LLM for consideration when generating codes. In this case, a stigma lens was selected to align with the content of Test Case Two.

Stigma within the world of mental health, drug use, addiction, and safe injection sites is extremely common and consistently confronted in studies within this sphere (Cruz et al., 2007; Yang et al., 2007). Stigma theory conceptualises stigma as a socially constructed process in which human differences are identified and imbued with negative meaning, leading to exclusion

and disadvantage. Erving Goffman describes stigma as an attribute that discredits individuals in a specific context, creating social rejection that is not inherent to the trait itself but arises from interaction and perception (1963). Link and Phelan (2001), building on this foundation, suggest that stigma occurs when labelling, stereotyping, separation into “us versus them”, status loss and discrimination co-occur within a context of power. Using stigma theory as an orienting theory for an LLM can work particularly well for coding data related to supervised injection site closures in Ontario, like the second dataset, since the discourse surrounding these sites is deeply shaped by moral judgment, power and social exclusion rather than neutral descriptions of health policy. In addition, using stigma theory provides a reflexive guardrail for the LLM by explicitly positioning drug use and harm reduction as socially contested rather than inherently problematic. This reduces the risk that the model simply reproduces dominant stigmatising frames present in the data and instead supports more analytically consistent coding of internalised stigma in narratives about SIS closures. In this way, stigma theory offers a theoretically grounded lens that aligns well with both qualitative health research and the interpretive demands of coding policy-relevant discourse.

When investigating the second dataset, it was observed that there was much more prominent sarcasm being expressed by users. This behaviour was identified during preliminary prompt testing, where generated codes reflected literal phrasing rather than the intended meaning of sarcastic statements. This content led to the first version of the prompt to generate codes with a meaning opposite to the input text in initial tests. In order to mitigate this effect, the prompt was instructed to consider the possibility of sarcasm or irony and “if the language contradicts its literal meaning, interpret intent appropriately”. This modification was intended to reduce misinterpretation of sarcastic content rather than to optimise overall coding performance; however, this may be a natural consequence of this modification.

Next, during the preliminary investigation of the dataset, it was observed that there was greater variation in the input text lengths compared to the first dataset. There were many input texts that were extremely short, leaving little content to interpret into codes. To test performance on extreme edge cases, these data points were not removed from the dataset. Retaining these cases allowed evaluation of model behaviour under sparse or underspecified inputs, which are common in real-world social media data. To give the LLM more information at the time of coding, the thread title was provided to the LLM in addition to the body of the comment and instructed to use the title as context. The title was used solely as contextual information and not as an independent source of input for coding. This approach was applied consistently across the dataset and was intended to support interpretation under sparse input conditions. Notably, this design choice also mirrors the methodology of the human coding study, in which coders were permitted to consult thread titles to inform their interpretations.

Lastly, the dimensions of the scoring framework were provided to the LLM as part of the specific guidelines to consider when coding. The LLM was instructed to “maintain consistency (follow a logical and uniform framing), relevance (each code clearly connects to the text),

distinction (codes capture unique ideas without redundancy) and specificity (avoid overly broad or vague codes)”. The complete version two prompt is given below:

```
prompt_template = ""
```

*You are a qualitative researcher performing thematic analysis inductively.  
This is a process of coding the data without trying to fit it into a preexisting coding frame, or  
the researcher’s analytic preconceptions*

*A thematic analysis usually consists of six steps:*

- 1. Familiarizing yourself with your data*
- 2. Generating initial codes*
- 3. Searching for themes*
- 4. Reviewing themes*
- 5. Defining and naming themes*
- 6. Producing the report*

*You were assigned to do step 2 on generating the initial codes.  
Another researcher already completed step 1 and after familiarizing themselves with the data  
they decided on the following research objective:*

*The goal of this study is to understand the online opinions and beliefs towards safe injection  
sites, by analyzing Reddit posts and comments from the r/Ontario subreddit.  
Analysing opinions of safe injection sites offers valuable insight into public perception that  
can be used to understand the themes underlying these discussions.  
The understanding of those themes could inform potential considerations by public health  
authorities surrounding safe injection sites.*

*For each comment, consider thread title to understand the context of the users comment, but  
focus on coding the comment itself.*

*thread title: {title}*

*comment: {text}*

*In order to complete Step 2 you need to follow these guidelines:*

- Focus on identifying basic segments or elements of the data that can be meaningfully analyzed.*
- Consider every possible theme or pattern during coding. Even if it seems unimportant at first, it might become relevant later.*
- Look for both explicit (semantic) and underlying (latent) content that stands out.*
- A comment may have multiple codes. Assign a maximum of 5 codes to each comment.*
- Include both positive and negative evaluations, as well as neutral or policy-oriented perspectives.*
- Avoid descriptive repetition—each code should add interpretive value.*
- Don’t ignore data that contradicts the dominant patterns. These can offer valuable insights.*

- Consider the possibility of **sarcasm or irony** — if the language contradicts its literal meaning, interpret intent appropriately.  
If a text is determined ironic or sarcastic, the polarity of the sentiment should be flipped. Do not just use a sarcastic/ironic label as a code.  
**consistency** follow a logical and uniform framing, **relevance** (each code clearly connects to the text), **distinction** (codes capture unique ideas without redundancy), and **specificity** (avoid overly broad or vague codes).  
- Incorporate a **stigma theory lens**: Based on Erving Goffman's stigma theory, note when comments reflect or challenge stigma related to drug use, users of safe injection sites, or harm reduction programs.

#### ### Output Instructions

Generate **a numbered list** of up to 5 items.  
Each item should follow this exact format: `1. **Code** - Explanation`  
Do not label the words "Code" or "Explanation" explicitly.  
Ensure each explanation briefly clarifies why the code fits the text.

Now, using the above framework, generate the codes for this comment: {text} “”

In addition to providing this context and instructions, three parameters were specified within the specific model configuration: temperature, seed and maximum token limit. Both seed and maximum token limit were held constant in this study, with temperature varied to create distinct LLM “agents”. The seed controls the random number generator used during generation; by fixing the seed to a specific value, researchers can reproduce the same output from the model across runs (OpenAI, 2025). The token limit determines the maximum number of tokens, which are equivalent to word parts, that the model can process in a single input-output sequence. This has implications for the length of input text that can be generated in one pass (IBM, 2024). Together, these parameters enable researchers to tailor LLM behaviour to explore different model behaviours and to fine-tune outputs for different analytic tasks. Temperature is associated with the so-called creativity of the output, where a smaller value, approaching zero, corresponds to a more deterministic, predictable response, whereas a larger value (the maximal value of 1.0) increases entropy, leading to randomness (Peeperkorn et al., 2024).

A maximum of five codes were generated for each data point, which was repeated with modified model parameters to create separate LLM “agents”. This decision was informed by an initial exploratory phase, during which the model was allowed to generate codes freely.

Additionally, in discussions with experienced qualitative researchers, it was noted that setting an average number of initial codes is sometimes used as a practical strategy in analysis, especially in datasets with multiple or novice coders (Creswell, 2007). Specifying a limit helps focus the coding process, avoiding over-fragmentation and simplifying subsequent analysis by ensuring the volume of generated codes remains manageable (Braun & Clarke, 2006). In addition to the generated codes, the LLM was asked to provide an explanation for each code that it generated. Each agent’s output was saved as a .csv file independently and merged into a single .csv file, which could then be used in the next step of initial code evaluation. This structure

supports systematic analysis of variability in code generation and can inform the selection of the most reliable configuration in subsequent work.

#### 2.4.1 Iterative Temperature Test

In experiments for Test Case One, little variation was observed among agents with varied temperatures of 0.1, 0.5 and 1 in terms of cosine similarity scores as well as human evaluation. In order to confirm this initial finding, an iterative temperature test for temperatures 0.1 - 1 was performed for Test Case Two. This includes generating and evaluating the entire dataset, using the CReDS framework (see Section 2.5.2 and 3.1), for 10 distinct temperatures from 0.1 to 1. An added benefit of this test is to reduce analytical load in the evaluation for Test Case Two, as the highest performing agent will be selected for all future tests and evaluations. The highest performing agent was selected according to the CReDS framework, where the highest scores in the most dimensions determined the highest performance.

#### 2.4.2 Seed Stability Test

In addition to the iterative temperature test, a seed stability test was performed using the highest-performing agent identified in the iterative temperature test. All other experimental conditions, including input texts, prompts and evaluation procedures, were held constant across runs. In prior work, it has been shown that LLMs produce consistent outputs and offer a promising approach to improving reliability and replicability in studies (Cui et al., 2025). To confirm replicability, a seed stability test was performed in which the seed was held constant to repeat code generation and evaluation ten times. Evaluation using the CReDS framework was compared for each run, one through ten, to determine the similarity of the outputs. In addition, a sample of the results was analysed manually to confirm alignment.

### 2.5 Data Filtering

#### 2.5.1 Test Case 1: Alberta Vaccine Passport Discourse

To ensure consistency and eliminate formatting discrepancies within the dataset, a multi-step filtering process was applied. As a preliminary step, any duplicated data points are removed. Twenty-eight duplicated data points were removed, leaving 386 remaining data points prior to data filtering and processing. Given the unstructured nature of the data, further entries that did not conform to predefined structural requirements were removed.

The first stage of the filtering process focused on verifying the code values within the dataset with the original output. Specifically, each instance of code was required to appear in a structured format consisting of a numerical identifier followed by a period, an optional space and enclosed between asterisks (\*\*Code\*\*). This standardisation ensured that only correctly formatted entries were retained for further analysis. The second stage of the filtering process focused on ensuring that the LLM code explanations were properly extracted from the original output. Given the unstructured nature of the dataset, inconsistencies in textual representation posed a risk to data integrity. These inconsistencies include generated codes or explanations that

were not included in the original output or formatting that interfered with subsequent processing steps. To address this, a filtering function was applied to retain only rows in which the LLM explanation appeared as a substring of the original output, regardless of case sensitivity. This approach ensured that the model's explanations were accurately captured in the output text, preventing the inclusion of entries that were incorrectly formatted or not part of the original output. The third stage of the filtering process aimed to enforce proper sentence structure for LLM code explanations by ensuring that all explanations concluded with a period. Given the variability in text formatting, inconsistencies such as missing punctuation could indicate cut-off sentences with incomplete meaning. The fourth stage of the filtering process focused on eliminating incomplete entries by removing rows containing blank values in any column. This process was repeated four times with filtering to further increase the number of data points. After all data filtering, the dataset had 370 data points remaining. This filtering results in the removal of 16 data points, representing 3.9% of the original non-duplicated dataset.

### 2.5.2 Test Case 2: Ontario Safe Injection Site Closure Discourse

Regarding data filtering for Test Case Two, the minimum number of data points was filtered out to maximise retention for analysis. This was done to assess the true performance of LLMs as assistive coders, including any hallucinations or poorly formatted output. Specific data points (5/645; 0.78%) were removed during evaluation due to missing data. In addition, when sampling data for the human evaluation stage, sampled data were only chosen out of a smaller (550 vs 645) dataset that had human-generated codes for comparison. This did not reduce the number of data points chosen for analysis, but did remove the possibility of selection for 95 data points.

## 2.6 Evaluation Techniques

Evaluation techniques for this study are split into two categories: quantitative evaluation and qualitative human evaluation. This dual approach enables a comprehensive assessment of the data and techniques employed. Through this process, methodological challenges in both the use of LLMs as assistive agents and the evaluation of generated codes can be identified and critically examined.

### 2.6.1 Initial Quantitative Evaluation of the LLM-generated Codes

All generated codes for Test Case One were evaluated using semantic similarity by utilising cosine similarity on the output of semantic embeddings. Cosine similarity is used to measure similarity by computing the cosine of the angle between the vectors; values close to 1 indicate parallel vectors and thus similar meanings, whereas values close to 0 indicate orthogonal vectors and thus dissimilar meanings. This is a common metric used and applied by a multitude of studies (De Paoli, 2024; Vrunda, 2024). The quantitative evaluation of the generated LLM codes compared the following metrics: text-LLM code similarity, text-explanation similarity and LLM code-original human code similarity. These calculated similarities were evaluated using

OpenAI's text-embedding-3-small to generate embeddings of the input text, codes and the explanation generated by the LLM. Determining the cosine similarity between the LLM code and the input text and the original human code can potentially estimate how well the LLM code can align with human codes. These evaluation techniques were applied to all 370 data points across all three agents.

## 2.6.2 Human Evaluation of LLM-generated Codes

### 2.6.2.1 Test Case 1: Alberta Vaccine Passport Discourse

The human evaluation of the generated codes began with five human coders generating new codes for the dataset. This was done as two years have passed since the original human coding was completed in 2023, and the context and perceptions around COVID-19 and mandatory vaccines may have shifted since that time. Experienced coders were recruited to re-code the entire dataset and were given the same prompt as the LLM to generate codes. This was done to maximise similarity between the methods of LLM code generation and human code generation.

Of the new dataset of human-generated codes, a representative sample was randomly selected to assess the intercoder reliability between LLM-generated codes and human-generated codes. A total of thirty-seven input texts were selected using Python's random sample method. This subsample represents ten percent of the dataset, a common guideline to follow for reliable sampling (O'Connor & Joffe, 2020; De Swert, 2012). The subsample was assessed by a team of six trained independent reviewers to determine whether the LLM codes matched the meaning captured by human coders. These reviewers were instructed on the project details and objectives and trained to accept or reject the similarity between the codes to a similar threshold. This training consisted of a one-hour group training, which outlined the study objectives and explained the process for determining similarity between codes. Reviewers were shown examples of matching and non-matching codes to build consistency among the team, and following the training, each reviewer was asked to evaluate five data points. Their initial decisions were reviewed in a brief follow-up to confirm understanding and alignment with the protocol.

Exactly matching codes (also referred to as common codes) between the newly human-generated and LLM-generated codes for each corresponding input text were pre-screened using cosine similarity to isolate and identify common codes between human and LLM groups. These common codes were verified by a human to be the same in meaning but with small variations in phrasing or grammar. For example, "Frustration with Unvaccinated Individuals" vs "frustration with the unvaccinated" are identified to be exactly matching codes. With the common codes isolated, the reviewers would mark Yes/No for whether an LLM-generated code was appropriate to the input text and whether the code matches the meaning of the human codes generated for that same input text. The judgments of each reviewer were recorded and counted to determine the proportions of codes that were exactly matching or matching meaning, as well as having

additional meaning or missing meaning, as compared to the human codes for the corresponding input text.

#### 2.6.2.2 Test Case 2: Ontario Safe Injection Site Closure Discourse

Human evaluation for Test Case Two followed nearly the exact same methods as Test Case One, save for a few changes. First, there was no human re-coding necessary, as there were human codes generated recently and provided to this study for use. A total of 165 input texts were selected by stratified sampling using Python’s random sampling method. This sample accounts for 30% of the original dataset, which is well within the guideline for sample selection (O’Connor & Joffe, 2020; De Swert, 2012). The input texts were grouped into seven word count bins, which ensures that the distribution of input texts would reflect the original distribution, not only for the purpose of equal distribution but so that there would not be a large proportion of single-word input texts by chance. The distribution comparisons of the original and sampled datasets are available as Figures i and ii in Appendix 1.

The subsample was reviewed by a team of trained reviewers, many of whom participated in the first human evaluation. A total of seven reviewers were recruited for Test Case Two’s human evaluation, one of whom was involved with generating the human codes for the second dataset. The reviewers were trained again to ensure alignment within the team.

#### 2.6.3 Exploratory Investigations into Data

Several targeted exploratory investigations were conducted to explore challenges and patterns identified during the evaluation. These targeted exploratory analyses offered greater insights into the limitations and opportunities associated with using LLMs as assistive coders and allowed for a closer examination of the sensitivity and robustness of how the CReDS framework evaluated LLM-generated codes.

##### 2.6.3.1 Duplicated Coding Investigation

Twenty-eight data points in the first dataset were duplicated and were inadvertently retained during dataset preparation. Rather than removing these duplicates post hoc, they were used as an opportunity to conduct a secondary analysis examining agreement among coders. These duplicate data points were also re-coded by the human coders during the human evaluation process, providing a direct comparison of replicability between human and LLM coders. The original and duplicated text and codes were compared using cosine similarity to determine how closely codes were reproduced by both GPT 4o-mini and human coders.

##### 2.6.3.2 Sarcasm Investigation

A sarcasm-focused exploratory analysis was conducted to assess the LLM’s ability to recognise and appropriately code instances of sarcasm. This analysis used data from the human evaluation process for the second dataset, which had much higher rates of sarcasm. In the human evaluation process, evaluators identified whether sarcasm was present in each input text and

whether either the human or LLM-generated codes were appropriate to the content. Within the subset of input texts flagged as containing sarcasm, it was determined if the LLM explicitly coded for sarcasm and was combined with the information of whether the human and LLM codes were judged appropriate to the text input. Crude counts and proportional distributions for these indicators were compiled and summarised in tabular form. In addition, a direct comparison of CReDS scores for input text with and without sarcasm was visualised. This analysis provided insight into the extent to which the LLM is able to detect sarcasm and maintain coding accuracy when confronted with texts characterised by ironic, witty or sarcastic language. In addition, a human comparison was included to determine whether challenges with sarcasm detection were unique to the LLM or to quantify the degree to which the LLM underperformed relative to human coders on this specific task.

#### 2.6.3.3 Specificity in Extreme Performers Analysis

An exploratory analysis into the specificity of LLM-generated codes was conducted to further examine qualitative differences between high and low-performing outputs. This analysis was not intended as a formal validation, but as a qualitative diagnostic to contextualise the specificity dimension of the CReDS framework.

From the human-evaluation subset, the thirty highest- and thirty lowest-scoring LLM codes were selected for detailed assessment. This analysis consisted of two components, an open-ended evaluation and a rubric evaluation. First, evaluators were asked to rate each code's overall specificity as specific (yes), not specific (no) or somewhat specific, acting as an open-ended evaluation. In addition, to reduce variation arising from individual interpretation, evaluators were asked to separately assess three elements of specificity via a rubric evaluation. They were asked to evaluate the following: whether the code is analytically useful, whether it is explicitly situated in context, and whether it is narrow and well-defined. Again, these codes are rated as yes, no, or somewhat. Secondly, evaluators conducted an in-depth qualitative examination of both groups of codes, responding to prompts such as: “What do the high-scoring codes do that the low-scoring codes do not?” “Are the low-scoring codes vague or overly generic? Are the high-scoring codes overly narrow or hyper-specific?” “Do the codes accurately reflect the input text?” “Are there shared characteristics in the input text associated with high or low scoring codes?” as well as an opportunity to list any other trends and patterns that they may see.

This analysis aimed to identify trends in specificity across the top- and bottom-scoring codes, offering insights into the qualitative attributes that distinguish strong LLM-generated codes from weaker ones. In addition, this offers insights into how accurately the CReDS framework is scoring the specificity of LLM-generated codes.

### 3. Results

#### 3.1 CReDS Framework

The CReDS framework is built around four key dimensions associated with “good quality” qualitative codes: Consistency, Relevance, Distinction and Specificity. These dimensions synthesise long-standing qualitative methodology principles with contemporary needs for evaluating machine-generated codes.

What makes a code “good quality”? In traditional qualitative research, good codes function to link raw data to interpretation (Charmaz, 2001). Across methodological traditions, “good quality” codes share several core attributes:

**Analytically useful:** High-quality codes facilitate interpretation. They help make sense of the data, organise concepts, identify patterns and build higher-order meaning. (Linneberg and Korsgaard, 2019; Eakin and Gladstone, 2020)

**Grounded and relevant:** High-quality codes are closely tied to the data itself, clearly emerging from the data rather than being imposed arbitrarily or excessively shaped by prior assumptions. (Korstjens and Moser, 2017; Seidel and Kelle, 1995)

**Specific and clear:** Codes should capture a discrete idea or phenomenon. They avoid overgeneralizations and articulate clearly (Coffey & Atkinson, 1996; Gibson & Brown, 2010). Specificity is especially important for preventing “catch-all” codes that obscure nuance.

**Consistent:** Similar data excerpts should receive the same or similar codes. This dimension underlies interpretive coherence and analytic transparency, both of which are important for reproducibility in computational approaches. (Lim, 2024, Lumivero, 2025; Mohammadi et al., 2025)

**Distinct:** Codes should be sufficiently differentiated from one another such that they do not overlap so much that they are redundant. Distinction ensures that each code has a unique analytic function and contributes meaningfully to the overall coding structure. (Arthur, 2021)

The selection of the four CReDS dimensions was guided by foundational principles shared across major qualitative research traditions. Although methodological approaches differ in their epistemological commitments, they converge on the expectation that high-quality codes must be coherent, grounded in the data, analytically meaningful and conceptually precise (Shen, 2025). These dimensions, therefore, represent qualities that experienced qualitative researchers routinely use, implicitly or explicitly, to assess the robustness of their coding structures. Importantly, these dimensions also lend themselves to operationalisation in computational contexts. Each can be translated into reproducible, semi-quantitative measures, making the CReDS framework suitable for evaluating both human and LLM-generated codes while maintaining fidelity to qualitative analytic values.

The CReDS framework is conceptually distinct from traditional intercoder agreement (ICA) metrics, or similar comparisons to human ground truths, which assess the extent to which multiple human coders apply the same code to the same text segments (Halpin, 2024). ICA measures coder alignment rather than code quality. In contrast, the CReDS framework evaluates the intrinsic qualities of the codes themselves; how well they fit the data, how precisely they capture distinct concepts and how consistently they function across a corpus. While ICA is useful for documenting procedural reliability, it offers little insight into whether the codes are analytically meaningful or theoretically sound. In addition, ICA can be logistically challenging and costly, limiting both scope and inclusivity of qualitative research (Zhao & Liu, 2025). The CReDS framework addresses this gap by focusing on conceptual rigour rather than coder agreement. This distinction is particularly important in the context of LLMs, where the question is not whether multiple coders agree with one another but whether the generated codes themselves meet accepted standards. To compound this, current evaluations of generated codes often rely heavily on comparison to human ground truth or computational metrics such as cosine similarity, accuracy or F1 scores (Katz et al., 2022; Dai et al., 2023; Chen et al., 2024; Raza et al., 2025; Hila & Hauser, 2025; Yi et al., 2025). While these approaches can provide useful descriptive information, they do not capture the interpretive nuance that qualitative research requires. From the perspective of qualitative methodology, the central question is not simply whether an LLM’s output matches a reference label, but whether the generated codes exhibit the elements that make a code meaningful within qualitative inquiry. When evaluating the feasibility of using LLMs as assistive coders, the generated codes must be evaluated with the considerations of the researchers using this tool and the traditions of the community at large.

**CONSISTENCY:** How consistent is the LLM in generating outputs? A similar input should get a similar output.

In order to evaluate consistency, a metric called Cosine Similarity Agreement (CSA) score was created and used to estimate the relationship between the similarity of text inputs and the similarity of text outputs. This score assumes that if the input texts are similar, the generated codes will exhibit an equivalent level of similarity. The mathematical relationship is shown in equation (1).

$$CSA = 1 - |S_{Text} - S_{Code}| \quad (1)$$

*Where  $S_{Text}$  (text similarity) = the cosine similarity calculated between the embeddings of the input text*

*Where  $S_{Code}$  (code similarity) = the cosine similarity calculated between the embeddings of the output codes*

*Implementation:* The CSA formula is a normalised similarity measure based on absolute difference. This type of formula is common across many disciplines, including mathematics and machine learning, psychology and psychometrics, ecology and many more. Within the evaluator Python code, each text input is compared against the corpus to find a similar text using cosine

similarity. Identifying these similar input texts, the codes of each text are then compared to determine the similarity of the respective codes. The similarities of the text and codes are then used as  $S_{Text}$  and  $S_{Code}$  within the CSA equation. The CSA score is saved as individual scores for each respective code.

*Interpretation:* If the input text and generated codes have close similarity values, the CSA score is interpreted as the generated output being consistent. More specifically, if the generated codes are similar and the input texts are similar, the CSA score will approach 1, meaning that the LLM was consistent. If the codes are similar but the input texts differ, the CSA score will approach 0, indicating that the LLM was inconsistent. This relationship is illustrated in Figure 3.

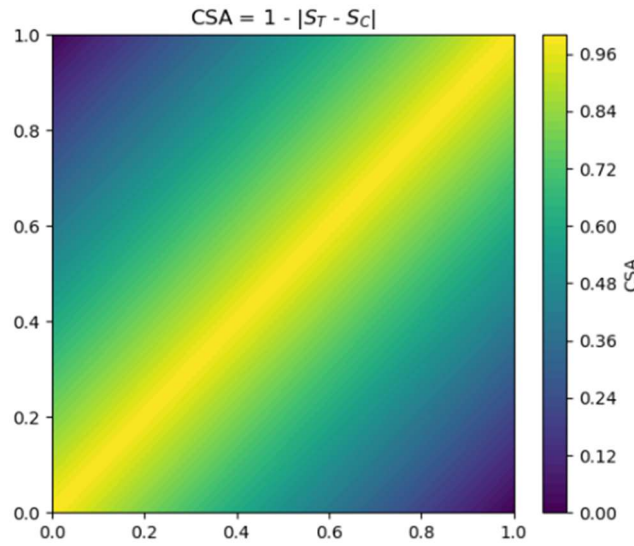


Figure 3 Heatmap of the CSA score equation

This graph shows the relationship between  $S_{Text}$  and  $S_{Code}$  as it relates to the CSA score. The heatmap illustrates that if  $S_{Text} \approx S_{Code}$ , the CSA score will approach 1, indicating high consistency of output.

**RELEVANCE:** How clearly is the code linked to the data? Is it grounded?

To evaluate Relevance, an LLM was provided with a rubric and instructed to evaluate each code according to the metrics specified in the rubric. This rubric, shown in Table 2 below, has four subdimensions in order to provide an overall score for relevance. These subdimensions are conceptual alignment, contextual alignment, topical alignment and semantic coherence; designed to capture distinct elements of relevance, including concept, context, specific topic and groundedness (Cosijn & Ingwersen, 2000).

*Implementation:* An LLM was utilised in order to evaluate each code's relevance to the input text. Using GPT-4o-mini, each individual code-text pair was provided to the LLM and evaluated according to the provided rubric. Each subdimension score was saved in addition to the overall relevance score.

Table 2 Evaluative Rubric for Relevance

| Metric                                   | 0                                                                                                    | 1                                                                                                                                                                     | 2                                                                                                                     | 3                                                                                                                         |
|------------------------------------------|------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>Conceptual alignment</i>              | The code has no meaningful connection to the main idea of the text or reflects an unrelated concept. | The code captures a peripheral or minor idea in the text, but misses the primary concept or misinterprets the text's intent.                                          | The code represents the main concept of the text but lacks depth or precision (e.g., too broad or missing key nuance) | The code directly reflects the main conceptual focus of the text, accurately summarizing its central meaning              |
| <i>Contextual alignment</i>              | The code conveys a tone, stance or evaluative meaning opposite to or unrelated to that of the text.  | The code partially matches the text's context or stance but distorts tone or intent (e.g., neutralising strong emotion or implying support when the text is critical) | The code aligns broadly with the text's tone and intent but omits subtle cues (e.g., irony, ambivalence)              | The code accurately mirrors the text's stance, tone, and intent, preserving evaluative direction                          |
| <i>Topical alignment</i>                 | The code refers to an entirely different topic or subject than the text                              | The code touches on a vaguely related theme or entity, but does not match the main subject of the text                                                                | The code and text discuss the same broad topic or phenomenon but differ in focus (e.g., same issue, different aspect) | The code and text clearly refer to the same specific topic, event or entity                                               |
| <i>Semantic Coherence (Groundedness)</i> | No overlap or meaningful connection in wording or meaning between code and text                      | Minimal lexical or semantic similarity; vague conceptual connection                                                                                                   | Some overlap in wording or clear use of synonyms/conceptually related terms, but partial or inconsistent              | Strong semantic linkage between code and text, shared key phrases, synonyms or paraphrased ideas appear clearly connected |

*Interpretation:* Each subdimension is scored between zero and three by the LLM and normalised between zero and one for interpretation. Scores closer to one are considered relevant to the input text, where scores approaching zero are considered not relevant to the input. In addition, the scores over the subdimensions are summed and normalised between zero and one as an overall score for relevance. These scores are interpreted the same way as the individual subdimension scores.

**DISTINCTION:** Are the codes redundant or clearly differentiated for an output? If an LLM generates five codes for a single output, how many will collapse into one?

Within each generated LLM code list, this dimension determines if any of the codes are similar enough to be considered redundant and how many of the original are distinct.

*Implementation:* The evaluator calculates the cosine similarity between the embeddings of the generated codes that are within a generated code list for each respective text input. Codes above a certain cosine similarity threshold ( $>0.85$ ) are labelled as redundant. This threshold was determined empirically through testing and manual inspection. The evaluator code identifies which codes are redundant for verification and assigns a distinctiveness score, which serves as the overall score for this dimension.

$$\text{Distinctiveness Score} = \frac{\# \text{ of non - redundant codes}}{\# \text{ of total codes}} \quad (2)$$

*Interpretation:* This dimension uses the calculated distinctiveness score as the overall score for distinction. A distinctiveness score of one indicates that no codes were considered redundant, where a distinctiveness score approaching zero indicates that all codes collapse into one single code (1/5 codes distinct).

SPECIFICITY: How narrowly does a code capture a concept?

To evaluate Specificity, an LLM was provided with a rubric and instructed to evaluate each code based on the metrics specified in the rubric. This rubric, shown in Table 3 below, has three subdimensions in order to provide an overall score for Specificity. These subdimensions are Scope, Context and Analytical Usefulness, designed to capture distinct elements of specificity (Saldana, 2016; Linneberg and Korsgaard, 2019; Eakin and Gladstone, 2020).

*Table 3 Evaluative Rubric for Specificity*

|                              | 0                                                               | 1                                                                | 2                                                   | 3                                                                              |
|------------------------------|-----------------------------------------------------------------|------------------------------------------------------------------|-----------------------------------------------------|--------------------------------------------------------------------------------|
| <i>Scope</i>                 | Overly broad, covers a very wide phenomenon with no boundaries. | General category, but lacks detail.                              | Reasonably precise and bounded.                     | Narrow and well-defined; tightly bounded by the data                           |
| <i>Context</i>               | No context provided                                             | Minimal context                                                  | Includes some context                               | Explicitly situates the data in context                                        |
| <i>Analytical usefulness</i> | Provides little to no analytical value                          | Limited analytical value; too general to guide theme development | Provides useful analytical power for theme building | Strong analytical value; captures a nuanced concept useful for theory building |

*Implementation:* An LLM was utilised in order to evaluate each code's specificity when compared to the input text. Using GPT-4o-mini, each individual code-text pair was provided to

the LLM and evaluated according to the provided rubric. Each subdimension score was saved in addition to the overall specificity score.

*Interpretation:* Each subdimension is scored between zero and three by the LLM and normalised between zero and one for interpretation. Scores closer to one are considered to be specific, where scores approaching zero are considered to be non-specific. In addition, the scores over the subdimensions are summed and normalised between zero and one as an overall score for specificity. These scores are interpreted the same way as the individual subdimension scores.

Each code is given a score between zero and one for each respective dimension. An overall score is not provided as it was found to erase the details of each respective dimension.

### 3.2 Test Case 1: Alberta Vaccine Passport Discourse

#### 3.2.1 Generated Codes

After filtering, 370 out of 386 data points were kept. Between the three agents, 5764 codes were generated with up to five codes per data point. There was a nearly even distribution between the agents, as demonstrated in Table 1. Among all agents' generated codes, 3221 (55.88%) were unique. Comparatively, there were 2077 (62.44%) codes generated by the group of new coders determined to be unique. As shown in Table 4 below, human coders generated more codes, which is likely due to the number of LLM-generated codes that were removed from filtering. Additionally, human coders generated slightly more unique codes as compared to LLM coders, highlighting the superficial nature LLM-generated codes can be vulnerable to.

*Table 4 Crude Generated Codes Count*

| <b><i>Evaluation</i></b>             | <b><i>Agent 1<br/>(T = 1.0)</i></b> | <b><i>Agent 2<br/>(T=0.1)</i></b> | <b><i>Agent 3<br/>(T=0.5)</i></b> | <b><i>Total</i></b> | <b><i>Human<br/>Coders</i></b> |
|--------------------------------------|-------------------------------------|-----------------------------------|-----------------------------------|---------------------|--------------------------------|
| <b><i>Number of Codes</i></b>        | 1905                                | 1925                              | 1934                              | 5764                | 2077                           |
| <b><i>Number of unique codes</i></b> | 1557                                | 1390                              | 1458                              | 3221                | 1297                           |

#### 3.2.2 Quantitative Evaluation of LLM-generated Codes

Cosine similarity was applied in three combinations on the codes generated for the input text compared to the original human-generated codes. First, by looking at the cosine similarity between the input text and code (Text- Code Similarity). Second, by comparing input text and code explanation (Text- Explanation Similarity), and lastly, by comparing LLM and original human code (Code - Original Code Similarity). Results can be found in Tables 5-6. A visualisation of the complete distributions is shown in Figures iii – v in Appendix 1 for each of the given metrics, with corresponding boxplots shown in Figures vi - viii.

Table 5 Summary of Cosine Similarities LLM Code to Original Code and Text Results,  $N = 370$

|                                             | <i>Evaluation Type</i>         | <i>Min</i> | <i>Mean</i> | <i>Median</i> | <i>Max</i> | <i>Q25</i> | <i>Q75</i> |
|---------------------------------------------|--------------------------------|------------|-------------|---------------|------------|------------|------------|
| <b>Agent 1</b><br><b><math>T=1</math></b>   | Text- Code Similarity          | 0.01       | 0.31        | 0.30          | 0.72       | 0.22       | 0.40       |
|                                             | Text- Explanation Similarity   | 0.14       | 0.48        | 0.48          | 0.77       | 0.41       | 0.55       |
|                                             | Code- Original Code Similarity | 0.04       | 0.29        | 0.27          | 0.93       | 0.20       | 0.37       |
| <b>Agent 2</b><br><b><math>T=0.1</math></b> | Text- Code Similarity          | 0.02       | 0.31        | 0.30          | 0.72       | 0.22       | 0.40       |
|                                             | Text- Explanation Similarity   | 0.09       | 0.48        | 0.48          | 0.76       | 0.42       | 0.55       |
|                                             | Code- Original Code Similarity | 0.02       | 0.29        | 0.27          | 0.93       | 0.20       | 0.36       |
| <b>Agent 3</b><br><b><math>T=0.5</math></b> | Text- Code Similarity          | -0.01      | 0.31        | 0.31          | 0.71       | 0.22       | 0.40       |
|                                             | Text- Explanation Similarity   | 0.14       | 0.48        | 0.48          | 0.76       | 0.42       | 0.55       |
|                                             | Code- Original Code Similarity | 0.04       | 0.29        | 0.27          | 0.93       | 0.20       | 0.37       |

Table 6 Summary of Cosine Similarities, Human Benchmark,  $N = 370$

| <i>Evaluation Type</i>                 | <i>Min</i> | <i>Mean</i> | <i>Median</i> | <i>Max</i> | <i>Q25</i> | <i>Q75</i> |
|----------------------------------------|------------|-------------|---------------|------------|------------|------------|
| <i>Text – Code Similarity</i>          | 0.06       | 0.35        | 0.35          | 0.75       | 0.26       | 0.43       |
| <i>Text – Explanation Similarity</i>   | 0.02       | 0.38        | 0.39          | 0.75       | 0.29       | 0.48       |
| <i>Code – Original Code Similarity</i> | 0.03       | 0.33        | 0.32          | 0.76       | 0.22       | 0.43       |

Table 5 shows highly consistent cosine similarities across all three LLM agents, with minimal variation across temperatures. For each evaluation type, the distributions and interquartile ranges are effectively stable across agents, suggesting that temperature differences do not meaningfully affect semantic alignment outcomes in this setting. When compared to human benchmarks, in Table 6, LLM agents exhibit similar distributional shapes across all similarity measures, highlighting alignment between LLM and human coders.

### 3.2.3 CReDS Evaluation

In order to provide further comparison for the second test case, the CReDS evaluation framework was utilised on the first test case for analysis. The results of each dimension are shown in Table 7 below, given for both LLM and human-generated codes. As observed in this table, there is strong alignment between human and LLM-generated codes.

*Table 7 Summary of CReDS Framework Evaluation, Test Case One N =370*

|                                  | <i>Consistency</i>   | <i>Relevance</i>     | <i>Distinction</i>   | <i>Specificity</i>   |
|----------------------------------|----------------------|----------------------|----------------------|----------------------|
| <i>Agent 2</i><br><i>T = 0.1</i> | <b>Min:</b> 0.56     | <b>Min:</b> 0.0      | <b>Min:</b> 0.56     | <b>Min:</b> 0.33     |
|                                  | <b>Med:</b> 0.86     | <b>Med:</b> 0.67     | <b>Med:</b> 1.00     | <b>Med:</b> 0.67     |
|                                  | <b>Max:</b> 1.0      | <b>Max:</b> 1.00     | <b>Max:</b> 1.00     | <b>Max:</b> 0.78     |
|                                  | <b>Mean:</b> 0.85    | <b>Mean:</b> 0.67    | <b>Mean:</b> 0.99    | <b>Mean:</b> 0.62    |
|                                  | <b>Std Dev:</b> 0.09 | <b>Std Dev:</b> 0.20 | <b>Std Dev:</b> 0.05 | <b>Std Dev:</b> 0.10 |
| <i>Human Coders</i>              | <b>Min:</b> 0.48     | <b>Min:</b> 0.0      | <b>Min:</b> 0.80     | <b>Min:</b> 0.33     |
|                                  | <b>Med:</b> 0.88     | <b>Med:</b> 0.67     | <b>Med:</b> 1.00     | <b>Med:</b> 0.67     |
|                                  | <b>Max:</b> 1.0      | <b>Max:</b> 1.0      | <b>Max:</b> 1.00     | <b>Max:</b> 0.89     |
|                                  | <b>Mean:</b> 0.86    | <b>Mean:</b> 0.60    | <b>Mean:</b> 0.98    | <b>Mean:</b> 0.98    |
|                                  | <b>Std Dev:</b> 0.09 | <b>Std Dev:</b> 0.23 | <b>Std Dev:</b> 0.05 | <b>Std Dev:</b> 0.09 |

Complementing Table 7, Figure 4 illustrates a substantial overlap between human and LLM-generated results across four evaluation dimensions. For nearly all dimensions, the LLM-generated codes score closely to human-generated codes, with comparable medians and overlapping ranges. Mean consistency scores are nearly identical between LLM and human coders (0.85 vs 0.86), as are relevance medians (0.67 for both groups) and distinction medians (1.0 for both groups), indicating broadly similar scores. Differences are most pronounced in specificity. The mean Specificity values are much higher for human coders than for LLMs (0.98 vs 0.62), and the median values are the same (0.67).

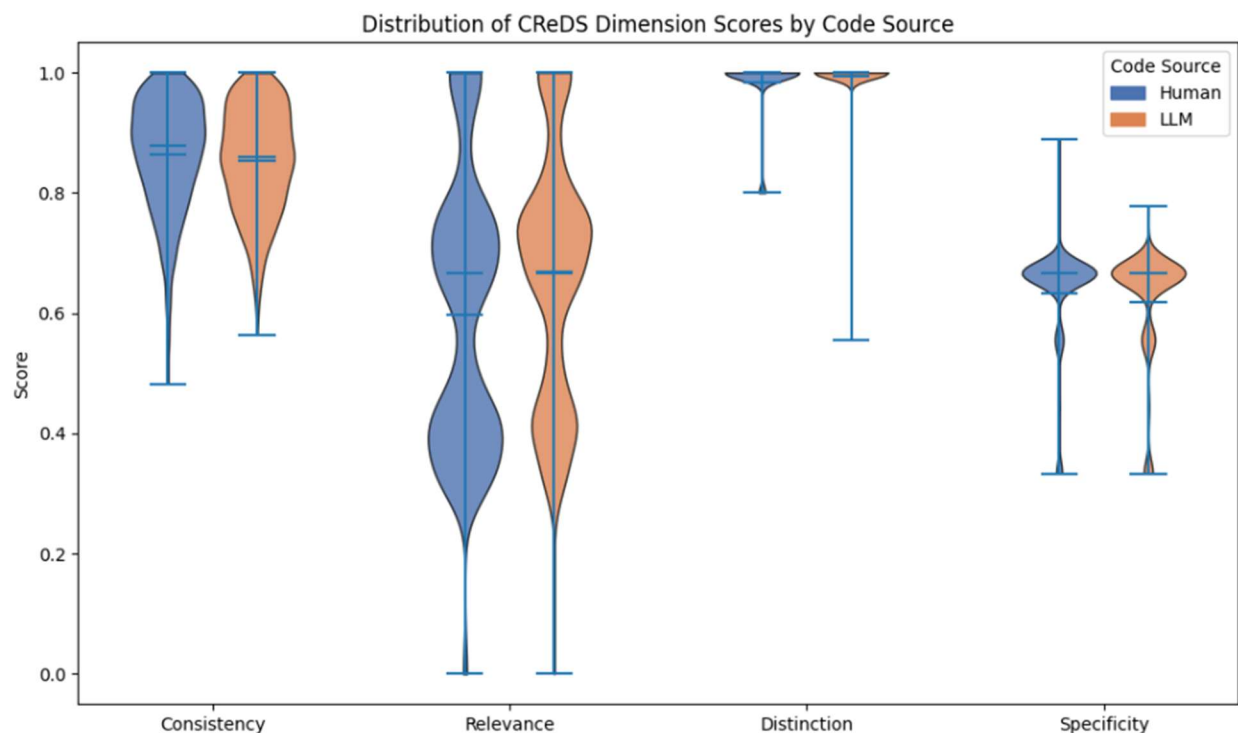


Figure 4 Distribution of CReDS Dimension Scores by Code Source, Test Case One

This figure shows strong alignment between human and LLM-generated codes across all dimensions. Specificity has the largest variation in mean (0.98 human vs 0.62 LLM), despite the median values being identical (0.67).

### 3.2.4 Human Evaluation of LLM-generated Codes

While quantitative metrics such as cosine similarity offer a useful approximation of textual alignment, they may not fully capture the nuance and contextual appropriateness required in qualitative coding. To address this, a qualitative human evaluation was conducted to further assess the quality and relevance of the LLM-generated codes. This step enables a more detailed examination of whether the LLM-generated codes were not only textually similar but also conceptually appropriate and meaningful within the context of the original input. Human evaluators independently reviewed each LLM-generated code and compared it against the corresponding input text and the original human-generated codes. The human evaluation captured both the degree of alignment and the LLM's capacity to identify novel but relevant codes. This provided insight into the potential to either reinforce or complement human coding efforts. The summary of the human evaluation as a percentage of the total codes per agent is summarised in Table 8. Performance is consistent across all three agents, indicating stable behaviour irrespective of agent configuration. These agents boast a 75.4-82.8% match between LLM and human codes, highlighting the success of the LLM-generated codes, not seen in the cosine similarity analysis seen in Tables 5-6 above. Of these matches, 22.4-23.6% were exact matches, showing modest alignment of LLM codes to human counterparts.

Table 8 Summary of Human Evaluation Percentages per Agent,  $N = 37 \times 3$

| %                | Total Codes Match | Code Match | Appropriate Text | Common Codes | Additional Information | Missing Information |
|------------------|-------------------|------------|------------------|--------------|------------------------|---------------------|
| Agent 1<br>T=1   | 82.76             | 59.20      | 67.82            | 23.56        | 13.79                  | 26.44               |
| Agent 2<br>T=0.1 | 75.43             | 53.14      | 64.14            | 22.29        | 17.71                  | 29.14               |
| Agent 3<br>T=0.5 | 79.88             | 57.47      | 65.52            | 22.41        | 16.67                  | 25.29               |

The total code match is the sum of the codes determined to match and the codes identified as exact matches (common codes) between the corresponding LLM and human-generated code lists. Codes that were found not to match the meaning of any of the codes in the respective human-generated group but were determined to be appropriate for the corresponding input text were defined as codes that identified additional information. Codes that were found not to match either the meaning of any of the corresponding human codes or the input text were determined to be missing information.

Codes that were determined not to match human meaning were potentially completely unrelated, for example, “Acknowledgement of Inconvenience” generated by an LLM agent, compared to “Necessary for entering businesses” generated by human coders. This could be due to the LLM-generated code being inappropriate to the input text, or alternatively, this code could be capturing appropriate meaning, but it simply does not match the meaning of the codes generated by human coders. Alternatively, a code could be determined not to match human meaning if it were too general and did not capture the specific meaning of the corresponding human codes. For example, in Table 9, the LLM-generated code “Healthcare Access” was determined not to match human coders for the input text shown, but also not to be appropriate to the input text. This LLM code, while relevant to the information in the input text, was determined to be too general to be identified as appropriate to the input text.

Table 9 Example of Non-Matching Codes

| Input Text                                                                                                                                      | Human Codes                                                                                                                                                                                                   | LLM Codes                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| vaccine passports<br>vaccine mandates<br>restrictions lockdowns<br>etc are all morally<br>clearer to me than<br>denying healthcare to<br>anyone | 'conditional support based on<br>fairness',<br>'supports restrictions like<br>lockdowns',<br>'against denial of healthcare',<br>'supports vaccine passports',<br>'questions morality of healthcare<br>denial' | <b>'Healthcare Access',</b><br>'Moral Clarity',<br>'Support for Vaccine Mandates',<br>'Support for Vaccine Passports',<br>'Critique of Healthcare Denial' |

These findings represent key considerations regarding the accuracy, relevance, and specificity of LLM-generated codes. The broader implications of these results, including their potential impact on the application of LLMs in qualitative research, will be examined in the subsequent discussion section.

### 3.3 Test Case 2: Ontario Safe Injection Site Closure Discourse

#### 3.3.1 Generated Codes

The entire second dataset was retained and coded, for a total of 645 data points. Codes were generated for ten agents, but all further results and analysis focus on a single agent for simplicity and due to the observations of the iterative temperature experiment. Table 10 illustrates the counts of LLM and human-generated codes, including the percentage of unique codes. For the agent selected, 3231 codes were generated, where 1901 (58.84%) were unique. In the human evaluation phase, 550 data points were used as the possible sampling pool to match the available human codebase.

*Table 10 Counts of LLM and Human-generated Codes*

|                               | <i>LLM (<math>T = 0.1</math>)</i> | <i>Human Coders</i> |
|-------------------------------|-----------------------------------|---------------------|
| <i>Number of Codes</i>        | 3231                              | 736                 |
| <i>Number of unique codes</i> | 1901                              | 44                  |
| <i>% of Codes Unique</i>      | 58.84                             | 5.98                |

#### 3.3.2 Iterative Temperature Test

Across the ten evaluated temperature settings (0.1–1.0), the distributions of CReDS scores were highly similar. For all four dimensions, Consistency, Relevance, Distinction and Specificity, score distributions showed minimal variation across temperatures, as shown in Figure 5. Minor fluctuations were observed at individual temperatures; however, these differences were small in magnitude and did not meaningfully alter the overall distributional patterns. Together, these results demonstrate substantial stability across temperature configurations, indicating that induced variations in temperature have minimal influence on the quality and accuracy of generated codes. This supports the findings from Test Case One and is consistent with prior work reporting minimal effects of temperature on code quality (Borchers et al., 2025). Furthermore, these findings justify the selection of a single temperature for use in subsequent experiments. Given the minimal impact of temperature on code quality, fixing the temperature reduces computational overhead and processing time without meaningfully affecting results. This further suggests that deploying multiple agents offers limited additional benefits in this specific context.

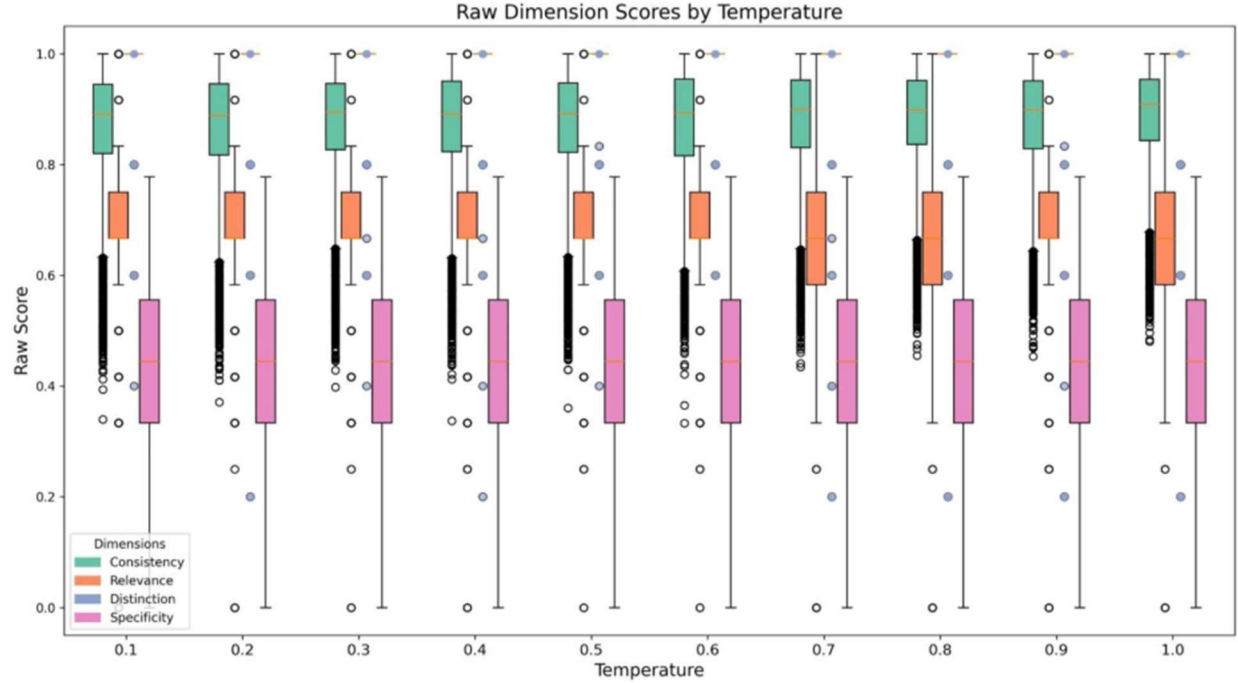


*Figure 5 Distribution of Dimension Scores per Temperature*  
This figure shows high consistency across temperature iterations, with all values nearly identical. The consistency dimension is consistently the lowest-scoring dimension, with a wide spread of values.

### 3.3.3 Seed Stability Test

Using the highest-performing agent identified in the iterative temperature test, a seed stability test was conducted across ten runs with a fixed seed. As with the results from the iterative temperature test, the evaluation results demonstrated a high degree of stability across runs. This is evident in Figure 6, which shows the stable nature of the seed stability test. For all four CReDS dimensions, the distributions of CReDS scores across the ten repetitions were nearly identical, with overlapping ranges and closely aligned central tendencies. Measures of the score distributions showed minimal variation between runs, with no degradation in performance observed in successive runs. This suggests that LLM-generated codes are highly reproducible under a fixed seed configuration.

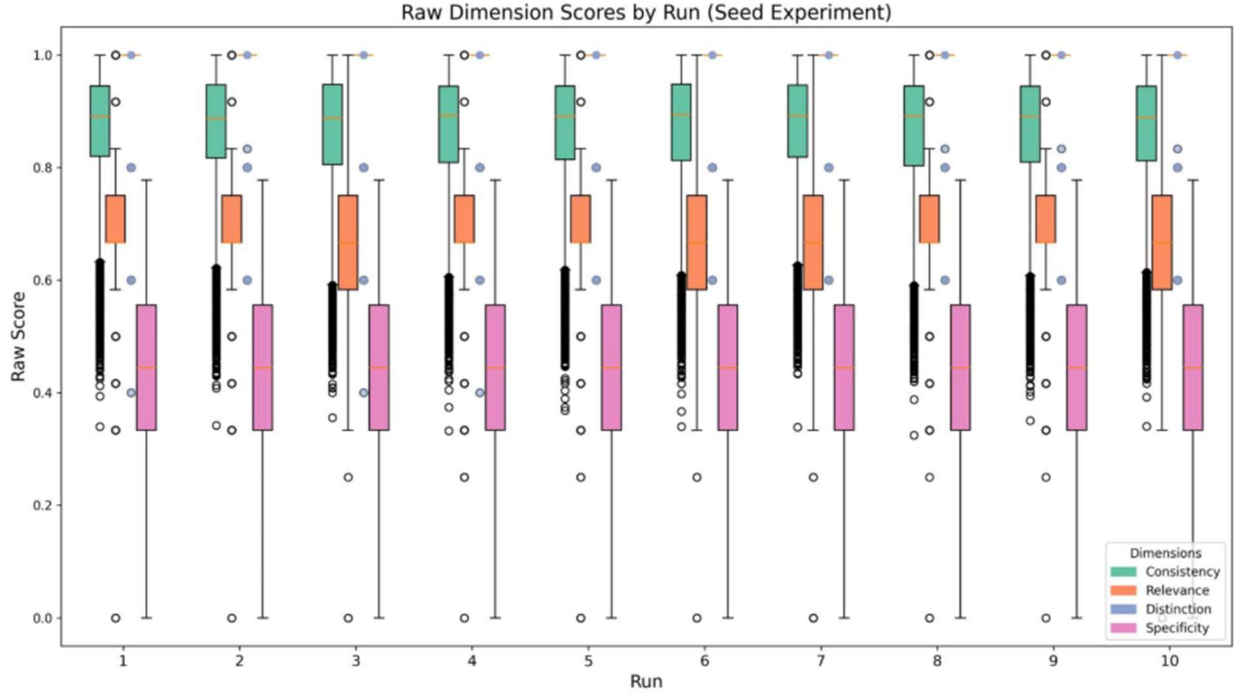


Figure 6 Distribution of Dimension Scores per Seed Run.

This figure shows extreme consistency among runs of the same seed, where all values for all dimensions are nearly identical. Mirroring the iterative temperature results, the consistency dimension is consistently the lowest-scoring dimension, with a wide spread of values.

### 3.3.4 CReDS Evaluation of LLM-generated Codes

Table 11 Summary of CReDS Framework Evaluation, Test Case Two

|                         | Consistency                                                                                             | Relevance                                                                                              | Distinction                                                                                            | Specificity                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| T = 0.1<br>N = 645      | <b>Min:</b> 0.58<br><b>Median:</b> 0.91<br><b>Max:</b> 1.0<br><b>Mean:</b> 0.90<br><b>Std dev:</b> 0.07 | <b>Min:</b> 0.0<br><b>Median:</b> 0.67<br><b>Max:</b> 1.0<br><b>Mean:</b> 0.62<br><b>Std dev:</b> 0.15 | <b>Min:</b> 0.40<br><b>Median:</b> 1.0<br><b>Max:</b> 1.0<br><b>Mean:</b> 0.98<br><b>Std dev:</b> 0.07 | <b>Min:</b> 0.0<br><b>Median:</b> 0.44<br><b>Max:</b> 0.78<br><b>Mean:</b> 0.43<br><b>Std dev:</b> 0.12 |
| Human Coders<br>N = 550 | <b>Min:</b> 0.49<br><b>Median:</b> 0.81<br><b>Max:</b> 1.0<br><b>Mean:</b> 0.80<br><b>Std dev:</b> 0.12 | <b>Min:</b> 0.0<br><b>Median:</b> 0.33<br><b>Max:</b> 1.0<br><b>Mean:</b> 0.32<br><b>Std dev:</b> 0.25 | <b>Min:</b> 0.18<br><b>Median:</b> 1.0<br><b>Max:</b> 1.0<br><b>Mean:</b> 0.98<br><b>Std dev:</b> 0.11 | <b>Min:</b> 0.0<br><b>Median:</b> 0.33<br><b>Max:</b> 0.78<br><b>Mean:</b> 0.43<br><b>Std dev:</b> 0.17 |

Table 11 above presents the CReDS evaluation results for Test Case Two, comparing LLM and human-generated codes. Overall, the LLM demonstrates higher scores than human coders for Consistency (median 0.91 vs 0.81) and Relevance (median 0.67 vs 0.33), where Distinction and Specificity medians are closely aligned between groups. Across dimensions,

Distinction shows the highest median and mean scores for both human and LLM-generated codes, suggesting that both coding approaches maintain conceptual distinctiveness. Human codes show greater variability, suggesting that the LLM may apply code more consistently, but potentially in a more generalised manner. These results are visualised in violin plots in Figure 7 below.

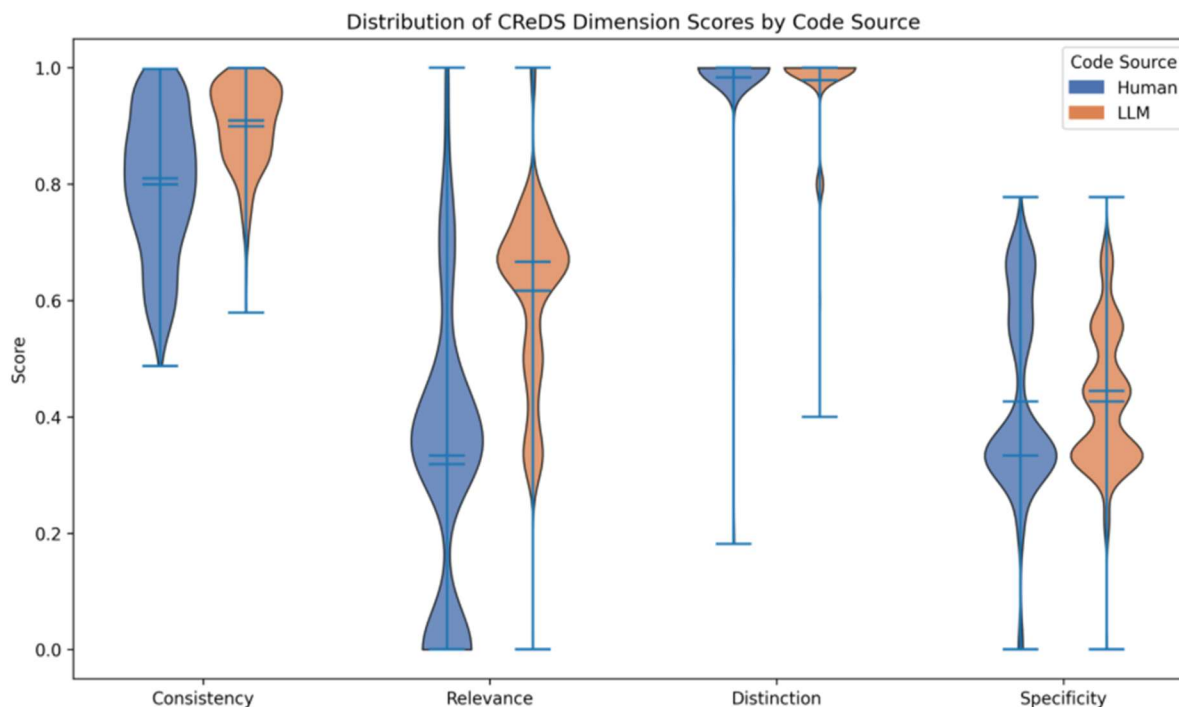


Figure 7 Distribution of CReDS Dimension Scores by Code Source, Test Case Two

This figure shows strong alignment between human and LLM codes, with some differences in the distribution of scores. The LLM code distributions appear to score slightly higher than human codes, notably in the Relevance dimension and Consistency, where the scores cluster at higher scores. This effect can be seen in reverse within the Specificity dimension for human codes, but at a smaller scale.

### 3.3.5 Human Evaluation of LLM-generated Codes

Table 12 summarises the alignment between LLM-generated codes and human coding for text-level inputs. The LLM code lists matches human codes in 71.6% of input texts, demonstrating a moderate degree of overlap. A higher proportion of LLM code lists were deemed appropriate to the text (95.7%), indicating that the majority of codes captured relevant meaning. The LLM had similar percentages of additional and missing information, suggesting that LLMs are well positioned to serve as assistive collaborators rather than independent agents.

*Table 12 Summary of Human Evaluation Percentages*

| N = 162    |       | LLM Code Match<br>Human | Appropriate to<br>Text | Additional<br>Information | Missing<br>Information |
|------------|-------|-------------------------|------------------------|---------------------------|------------------------|
| Text Level | Count | 116                     | 155                    | 120                       | 117                    |
|            | %     | 71.6                    | 95.7                   | 74.1                      | 72.2                   |

### 3.4 Probing Analyses

#### 3.4.1 Duplicated Coding Analysis

In examining the cosine similarity between human-human and LLM-LLM duplicates, codes were determined to be more similar for LLM-generated codes compared to human-generated codes. This relationship can be observed in Tables 13 and 14, where Table 14 shows specific examples from the dataset. Between all agents, the human-human cosine similarity median was 0.2 compared to 0.43-0.62 for LLM-LLM comparison. This finding aligns with prior research emphasising the challenges of achieving high intercoder reliability among traditional human coders (O'Connor & Joffe, 2020; De Swert, 2012; Neuendorf, 2017).

Table 13 Summary of Duplicate Analysis,  $N = 28$

| Agent              |         | LLM Duplicate Cosine Similarity | Human Duplicate Cosine Similarity | LLM-Human Duplicate Cosine Similarity Version 1-1 | LLM-Human Duplicate Cosine Similarity Version 2-2 | LLM-Human Duplicate Cosine Similarity Version 1-2 |
|--------------------|---------|---------------------------------|-----------------------------------|---------------------------------------------------|---------------------------------------------------|---------------------------------------------------|
| Agent 1<br>T= 1    | Min     | 0.40                            | 0.19                              | 0.08                                              | 0.06                                              | 0.13                                              |
|                    | Mean    | 0.45                            | 0.20                              | 0.16                                              | 0.13                                              | 0.16                                              |
|                    | Max     | 1.0                             | 0.20                              | 0.23                                              | 0.20                                              | 0.18                                              |
|                    | Median  | 0.45                            | 0.20                              | 0.16                                              | 0.13                                              | 0.16                                              |
|                    | Std dev | 0.04                            | 0.01                              | 0.08                                              | 0.07                                              | 0.03                                              |
| Agent 2<br>T= 0.1  | Min     | 0.43                            | 0.19                              | 0.02                                              | 0.13                                              | 0.12                                              |
|                    | Mean    | 0.57                            | 0.20                              | 0.09                                              | 0.14                                              | 0.14                                              |
|                    | Max     | 1.0                             | 0.20                              | 0.20                                              | 0.16                                              | 0.17                                              |
|                    | Median  | 0.43                            | 0.20                              | 0.02                                              | 0.13                                              | 0.12                                              |
|                    | Std dev | 0.19                            | 0.01                              | 0.09                                              | 0.02                                              | 0.03                                              |
| Agent 3<br>T = 0.5 | Min     | 0.62                            | 0.19                              | 0.10                                              | 0.16                                              | 0.13                                              |
|                    | Mean    | 0.77                            | 0.20                              | 0.13                                              | 0.16                                              | 0.14                                              |
|                    | Max     | 1.0                             | 0.20                              | 0.19                                              | 0.16                                              | 0.16                                              |
|                    | Median  | 0.62                            | 0.20                              | 0.10                                              | 0.16                                              | 0.13                                              |
|                    | Std dev | 0.19                            | 0.01                              | 0.05                                              | 0.004                                             | 0.02                                              |

*Table 14 Example of Duplicate Codes for Human and LLM-generated Codes*

| Input text: “across the country as a whole, most likely yes in alberta where we skew strongly conservative i doubt that profile would hold up look at it like this if the majority of ant vaxxers were liberal voters do you think that kenney would hesitate to hit them with a vaccine passport for even a minute” |                                                                                                                                                                        |                                                                                                                                                                                |                                                                                                                                                                                                            |                                                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Temp                                                                                                                                                                                                                                                                                                                 | Human Codes 1                                                                                                                                                          | Human Codes 2                                                                                                                                                                  | LLM Codes 1                                                                                                                                                                                                | LLM Codes 2                                                                                                                                                                                         |
| T= 0.1                                                                                                                                                                                                                                                                                                               | government is self-serving, government caters to anti-vaxxers, government opposes vaccine passports, politics affect policy, concerned with disappointment over health | Political views matter, hesitancy from government, concerned that politics affect policy, political spectrum predicts vaccine support, believes majority would accept passport | Perception of National Trends, Political Ideology Influence, Skepticism Towards Leadership, Antivaccine Sentiment and Political Affiliation, Hypothetical Scenario, Cynicism Towards Policy Implementation | Perception of National Trends, Political Affiliation and Vaccination, Skepticism Towards Leadership, Antivaccine Sentiment, Vaccine Passport Controversy, Regional Differences in Vaccine Attitudes |
| T = 0.5                                                                                                                                                                                                                                                                                                              | government is self-serving, government caters to anti-vaxxers, government opposes vaccine passports, politics affect policy, concerned with disappointment over health | Political views matter, hesitancy from government, concerned that politics affect policy, political spectrum predicts vaccine support, believes majority would accept passport | Perception of National Trends, Regional Political Influence, Skepticism Towards Government Action, Hypothetical Scenario, Critique of Leadership, Vaccine Passport Discussion                              | Perception of National Trends, Regional Political Influence, Skepticism Towards Government Action, Antivaxxer Demographics, Political Leadership and Policy Decisions, Hypothetical Scenario        |
| T = 1                                                                                                                                                                                                                                                                                                                | government is self-serving, government caters to anti-vaxxers, government opposes vaccine passports, politics affect policy, concerned with disappointment over health | Political views matter, hesitancy from government, concerned that politics affect policy, political spectrum predicts vaccine support, believes majority would accept passport | Perception of Political Influence, Skepticism of Vaccine Policies, Antivaccine Sentiment, Critique of Political Leadership, Regional Variation in Opinions, Cynicism towards Government Actions            | Perception of Political Influence, Skepticism of Local Trends, Antivaxxer Demographics, Government Response to Public Sentiment, Challenge to Assumptions, Critical Analysis of Leadership          |

### 3.4.2 Sarcasm Analysis

Table 15 reports several elements of codes generated from input texts that were explicitly flagged as containing sarcasm. How many codes were explicitly coded for sarcasm, how many codes (both LLM and human) were appropriate to the input texts, and whether the LLM code was inappropriate due to sarcasm were reported as counts and percentages of the respective totals.

*Table 15 Summary of Sarcasm Analysis*

|                          |                                                     |                                          |                                                       |                                            |
|--------------------------|-----------------------------------------------------|------------------------------------------|-------------------------------------------------------|--------------------------------------------|
| NLLM= 175<br>NHuman = 66 | <i>How many LLM codes are explicit for sarcasm?</i> | <i>LLM codes appropriate to the text</i> | <i>Was the LLM code inappropriate due to sarcasm?</i> | <i>Human codes appropriate to the text</i> |
| <i>Count</i>             | 31                                                  | 115                                      | 9                                                     | 49                                         |
| <i>Percentage</i>        | 17.7                                                | 65.7                                     | 5.1                                                   | 74.2                                       |

Within this analysis, it was also determined why a code may be inappropriate for the input text if it was not due to sarcasm. The counts and percentages for these reasons were reported in Table 16 below. The major reason for a code not being appropriate to the text was incorrect content. For example, for the text “pretty much nice one ontario,” one of the LLM-generated codes that was determined not to be appropriate due to incorrect content was the code “stigmatization of drug users”. Other reasons for inaccuracy included focusing on one small detail of the input rather than the real overall meaning, being too general to be analytically useful, direct labelling of sarcasm (ex. “Sarcasm”) and one input that was classified as “challenging”. This particular input was difficult even for human coders to decipher, as it had several unrelated concepts in quick succession.

*Table 16 Reasons for Code Inappropriateness, N=52*

|                         |                                                  |
|-------------------------|--------------------------------------------------|
| <i>Count/Percentage</i> | <i>Reason</i>                                    |
| 35 (67.3%)              | Incorrect content                                |
| 8 (15.4%)               | Focus on one small detail of the input           |
| 7 (13.5%)               | Too general to be analytically useful            |
| 2 (3.8%)                | Other (Direct labelling, challenging input text) |

As part of the targeted sarcasm analysis, individual codes were investigated. Table 17 presents representative examples of sarcastic input text alongside the corresponding LLM and human-generated codes. These examples were selected to illustrate how codes generated in the context of sarcasm by the model compared to human coders.

Table 17 Examples of Sarcastic Input Text and Corresponding Codes

| <i>Text</i>                                                                                | <i>LLM Code</i>                    | <i>Appropriate-ness</i> | <i>Human Code</i>                                            | <i>Appropriate-ness</i> |
|--------------------------------------------------------------------------------------------|------------------------------------|-------------------------|--------------------------------------------------------------|-------------------------|
| interesting ford speaks with such a depth of knowledge about drug dealers gee i wonder why | Skepticism of Authority            | Y                       | Personal attacks on Premier Doug Ford - Ford personal attack | Y                       |
|                                                                                            | Stigma of Drug Dealers             | N                       |                                                              |                         |
| so ford wants more open street drug use then                                               | Perceived Neglect of Public Safety | Y                       | Support for SIS closures                                     | N                       |
|                                                                                            | Stigmatization of Drug Users       | N                       |                                                              |                         |

As shown in Table 17, LLM-generated codes tend to capture the general orientation of sarcastic statements but can differ from human codes in terms of specificity and contextual grounding. LLMs produced broader codes that reflected the sentiment of the text but did not consistently identify the implicit irony embedded in sarcastic language. This was particularly evident when sarcasm relied on indirect critique or rhetorical implication, where the model’s output shifted toward generalized themes rather than the specific ironic target identified by human coders. These examples align with the findings from the sarcasm analysis, which showed that the most common reason for inappropriate outputs from sarcastic inputs is incorrect content rather than incorrect sentiment. These findings were not exclusive to LLM-generated codes, as it was found that some human-generated codes were not appropriate to the input text due to incorrect content, as shown in Table 17, however at a lesser rate than that of LLM.

Figures 8 and 9 display the CReDS dimension scores for the four dimensions separated by sarcasm label, using violin plots for human and LLM-generated codes. Both graphs show minimal differences between sarcastic and non-sarcastic inputs overall. In the LLM results, minor deviations are observed: the relevance violin for non-sarcastic inputs exhibits a longer tail than the sarcastic inputs, likely influenced by codes explicitly coding for sarcasm (ex, “Use of sarcasm”, “Sarcastic critique of authority”). Consistent with this visual pattern, the difference in mean CReDS scores for LLM-generated codes is -0.1% between sarcastic and non-sarcastic text inputs, indicating that the overall effect is minimal. However, the specificity violin for non-sarcastic inputs shows a higher peak, suggesting that sarcastic inputs may reduce the specificity of codes generated by LLMs. Conversely, in the human coder results, the non-sarcastic relevance violin has a higher peak, indicating that sarcasm may also negatively affect human coders’ ability to generate codes closely aligned with the input text. Overall, these results suggest that while the impact of sarcasm is minimal, it may affect certain dimensions for both LLMs and human coders.

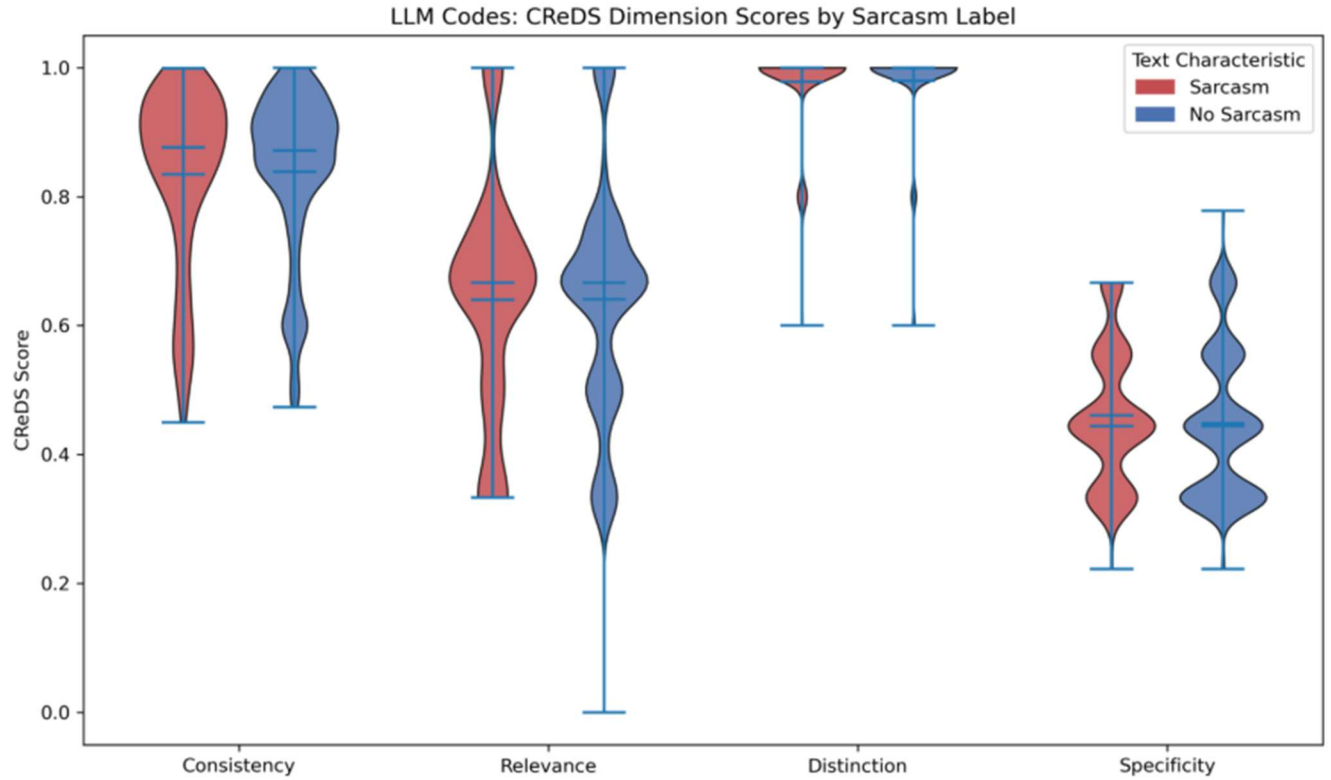


Figure 8 CReDS Dimension Scores by Sarcasm Label, LLM

This figure shows little impact from sarcastic and non-sarcastic input text on the CReDS scores of LLM-generated codes. There are some small variations in the distributions of scores, notably Relevance. In the Relevance dimension, the non-sarcastic text inputs have a longer tail showing higher variation than the sarcastic text counterparts. Additionally, it is observed that the Specificity dimension has a higher peak for non-sarcastic text inputs as compared to the sarcastic counterparts. Notably, when comparing LLM results from Figure 6 and human results from Figure 7, it is observed that both sarcastic and non-sarcastic results for Relevance have lower scores for human coders.

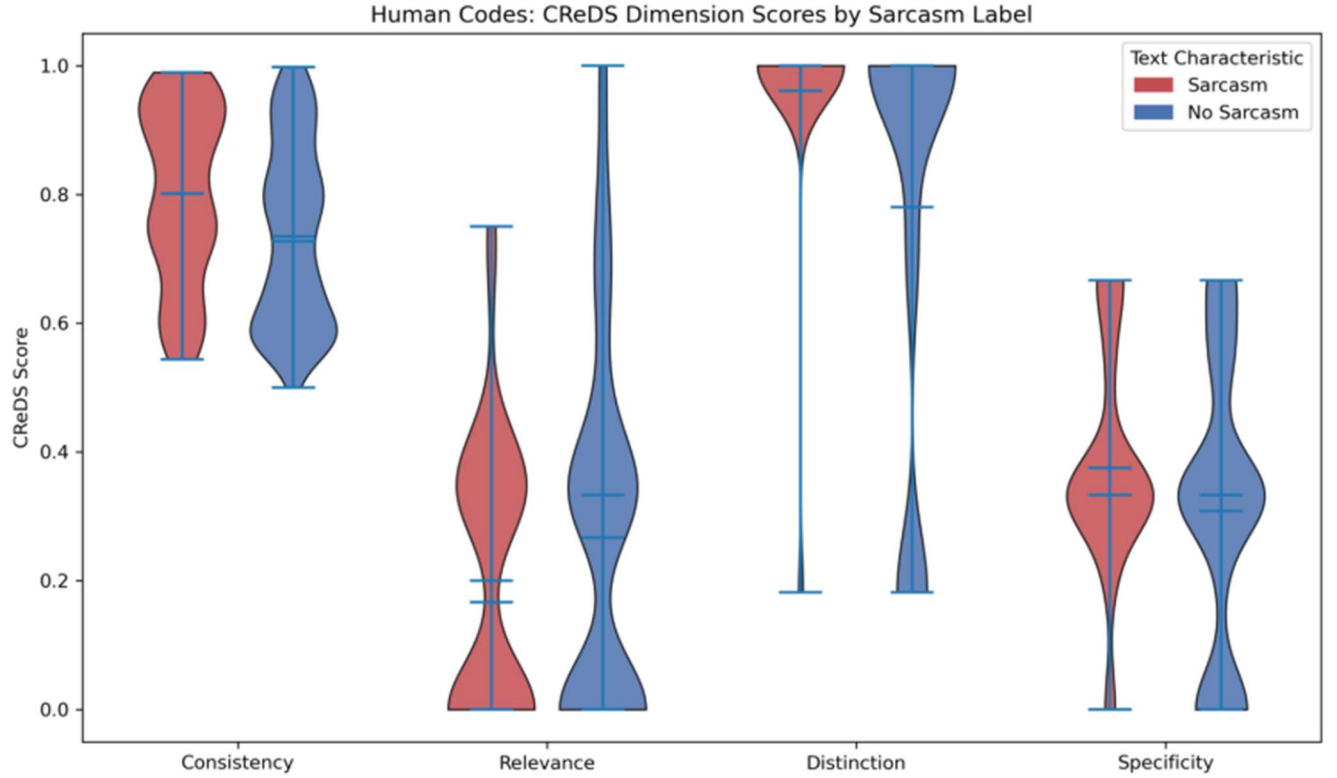


Figure 9 CReDS Scores by Sarcasm Label for Human-Generated Codes

This figure shows minimal impact from sarcastic and non-sarcastic input text on the CReDS scores of human-generated codes. There are some small differences in the distributions of codes, notably for the Relevance dimension, where there is a higher peak for the score for non-sarcastic text inputs.

Table 18 Percent Differences Between Sarcastic and Non-Sarcastic CReDS Scores Among Coder Groups

| %     | Difference | Consistency | Relevance | Distinction | Specificity |
|-------|------------|-------------|-----------|-------------|-------------|
| Human | Mean       | 9.0         | -25.1     | 23.1        | 21.5        |
|       | Median     | 10.1        | -50       | 0.0         | 0.0         |
| LLM   | Mean       | -0.5        | -0.1      | -0.2        | -8.6        |
|       | Median     | 0.5         | 0.0       | 0.0         | 0.0         |

Tables 18 and 19 present the CReDS scores and percent differences between sarcastic and non-sarcastic inputs for human- and LLM-generated codes. Table 18 shows the percent differences between sarcastic and non-sarcastic CReDS scores among coder groups, illustrating that among human-generated codes, the CReDS scores have much higher differences between sarcastic and non-sarcastic scores as compared to LLM-generated codes. Most notably in the Relevance dimension, human-generated code CReDS mean scores are 50% lower for sarcastic input texts, whereas LLM-generated code CReDS mean scores are 0.1% lower for sarcastic input texts. Table 19 further highlights differences between human- and LLM-generated codes for sarcastic and non-sarcastic inputs, showing markedly greater differences in the Relevance dimension. Specifically, the mean CReDS score for LLM-generated codes exceeds that of

human-generated codes by 220% for sarcastic input texts, underscoring a pronounced discrepancy in how relevance is captured under conditions of sarcasm.

*Table 19 Dimension Specific Differences Between Sarcastic and Non-Sarcastic CReDS Scores, LLM - Human*

|                                 |                      | <i>Consistency</i>      | <i>Relevance</i>        | <i>Distinction</i>     | <i>Specificity</i>      |
|---------------------------------|----------------------|-------------------------|-------------------------|------------------------|-------------------------|
| <i>Human</i>                    | <i>Sarcastic</i>     | Med: 0.80<br>Mean: 0.80 | Med: 0.17<br>Mean: 0.20 | Med: 1.0<br>Mean: 0.96 | Med: 0.33<br>Mean: 0.38 |
|                                 | <i>Non-sarcastic</i> | Med: 0.73<br>Mean: 0.74 | Med: 0.33<br>Mean: 0.27 | Med: 1.0<br>Mean: 0.78 | Med: 0.33<br>Mean: 0.31 |
| <i>LLM</i>                      | <i>Sarcastic</i>     | Med: 0.88<br>Mean: 0.83 | Med: 0.67<br>Mean: 0.64 | Med: 1.0<br>Mean: 0.98 | Med: 0.44<br>Mean: 0.46 |
|                                 | <i>Non-sarcastic</i> | Med: 0.87<br>Mean: 0.84 | Med: 0.67<br>Mean: 0.64 | Med: 1.0<br>Mean: 0.98 | Med: 0.44<br>Mean: 0.45 |
| <i>Human-LLM Difference</i>     | <i>Sarcastic</i>     | Med: 0.08<br>Mean: 0.03 | Med: 0.50<br>Mean: 0.44 | Med: 0.0<br>Mean: 0.02 | Med: 0.11<br>Mean: 0.09 |
|                                 | <i>Non-sarcastic</i> | Med: 0.14<br>Mean: 0.1  | Med: 0.33<br>Mean: 0.38 | Med: 0.0<br>Mean: 0.20 | Med: 0.11<br>Mean: 0.14 |
| <i>Human-LLM Difference (%)</i> | <i>Sarcastic</i>     | Med: 9.4<br>Mean: 4.2   | Med: 300<br>Mean: 220   | Med: 0.0<br>Mean: 1.8  | Med: 33.3<br>Mean: 23.0 |
|                                 | <i>Non-sarcastic</i> | Med: 19.8<br>Mean: 14.1 | Med: 100<br>Mean: 140   | Med: 0.0<br>Mean: 25.5 | Med: 33.3<br>Mean: 45.3 |

### 3.4.3 Specificity Analysis

Table 20 presents the CReDS specificity scores for the highest- and lowest-scoring code sets. This table also shows the human rater’s judgements of specificity, as both an open-ended judgement and as a defined rubric judgement. Among the bottom 30 codes, 57% were deemed appropriate, with a mean CReDS score of 0.29 and substantial variability across open-ended and rubric-based human judgements. In contrast, 80% of the top 30 codes were deemed appropriate, with higher and more consistent mean CReDS scores of 0.67, indicating that higher-performing codes were more reliably aligned with the input text. Overall, the table highlights that codes with higher CReDS scores of Specificity tend to correspond to higher appropriateness and more consistent human evaluations.

Table 20 Summary of Specificity Analysis, N = 60

| N = 60           | Was the code deemed appropriate? (count/%) |          | CReDS Score                                                                                           | Human Score                                                                                  |                                                                                              |
|------------------|--------------------------------------------|----------|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|                  | Yes                                        | No       |                                                                                                       | Open-Ended Judgement                                                                         | Rubric Judgement                                                                             |
| Bottom 30 Scores | 17 (57%)                                   | 13 (44%) | <b>Min:</b> 0.22<br><b>Med:</b> 0.33<br><b>Max:</b> 0.33<br><b>Mean:</b> 0.29<br><b>Std Dev:</b> 0.06 | <b>Min:</b> 0<br><b>Med:</b> 0<br><b>Max:</b> 1<br><b>Mean:</b> 0.3<br><b>Std Dev:</b> 0.39  | <b>Min:</b> 0<br><b>Med:</b> 1<br><b>Max:</b> 1<br><b>Mean:</b> 0.73<br><b>Std Dev:</b> 0.37 |
| Top 30 Scores    | 24 (80%)                                   | 6 (20%)  | <b>Min:</b> 0.67<br><b>Med:</b> 0.67<br><b>Max:</b> 0.78<br><b>Mean:</b> 0.67<br><b>Std Dev:</b> 0.02 | <b>Min:</b> 0<br><b>Med:</b> 1<br><b>Max:</b> 1<br><b>Mean:</b> 0.73<br><b>Std Dev:</b> 0.37 | <b>Min:</b> 0<br><b>Med:</b> 0<br><b>Max:</b> 1<br><b>Mean:</b> 0.30<br><b>Std Dev:</b> 0.39 |

Looking specifically at the top and bottom performing code for specificity, Table 21 below shows the input text, applied code and the subdimension and the overall scores of specificity. These codes show the wide variability of the abilities of LLM codes to generate specific codes.

Table 21 Top and Bottom Scoring Codes within Specificity Dimension

| Input Text                                                                                                                                                                                                              | Code                         | Score |         |                       |         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-------|---------|-----------------------|---------|
|                                                                                                                                                                                                                         |                              | Scope | Context | Analytical Usefulness | Overall |
| good news time to ship all the hobos to the arctic                                                                                                                                                                      | Dehumanization of Drug Users | 2     | 2       | 3                     | 0.78    |
| you want to hide homelessness and addicts away the easiest and cheapest way is universal basic income ubi under the program doug ford killed a person paid per month addict or not was housed and not committing crimes | UBI (Universal basic income) | 1     | 0       | 1                     | 0.22    |

As part of the evaluation, the differences in scoring between the top and bottom scoring codes were determined for each independent sub-dimension for both the CReDS evaluation and the human judgments. These results are shown in Table 22 below. This investigation found that there were much higher differences between CReDS evaluation of the top and bottom scoring codes as compared to human evaluations, indicating the possibility of greater sensitivity to code quality by CReDS scoring. In contrast, differences in human evaluations are smaller and more variable, with medians often close to zero, suggesting that human judgements are less sharply differentiated between high and low scoring codes.

Table 22 Difference in Scores in Top and Bottom Codes by Evaluator, N = 60

| Evaluation |         | Analytical<br>Usefulness | Context | Scope | Overall |
|------------|---------|--------------------------|---------|-------|---------|
| CReDS      | Min     | 0.33                     | 0.67    | 0.33  | 0.45    |
|            | Med     | 0.33                     | 0.33    | 0.33  | 0.33    |
|            | Max     | 0.67                     | 0.33    | 0.33  | 0.44    |
|            | Mean    | 0.34                     | 0.47    | 0.33  | 0.38    |
|            | Std Dev | 0.06                     | 0.17    | 0     | 0.04    |
| Human      | Min     | 0                        | 0       | 0     | 0.17    |
|            | Med     | 0.5                      | 0       | 0.5   | 0.33    |
|            | Max     | 0                        | 0       | 0     | 0       |
|            | Mean    | 0.37                     | 0.05    | 0.43  | 0.28    |
|            | Std Dev | 0.001                    | 0.03    | 0.09  | 0.03    |

When researchers were asked to evaluate the top and bottom 30 codes using targeted questions, several patterns were identified. These patterns were regarding the commonalities of the code quality and input texts for the respective groups. Researchers stated that the high-scoring codes were more specific to the content of the text, even using descriptive or specialised language to narrow the topic. Lower-scoring words were “vague and often describe a larger concept rather than a specific topic”. The researchers noted that it appeared lower-scoring codes had input texts that, on average, were shorter and many had “colloquial terms like 'bizzarro' or had inputs that were off-topic from the thread title.”

## 4. Discussion

This study highlights the potential feasibility of utilising LLMs as coders in qualitative research workflows, through the evaluation of GPT-4o mini, to serve as assistive agents in inductive qualitative research. The evidence shows that these models have the potential to generate contextually appropriate and methodologically consistent codes that often align with codes generated previously by humans. They are, however, not without limitations. Their use raises important methodological questions regarding implementation, reflexivity and the epistemic status of machine-generated codes. The discussion below explores these limitations within LLM qualitative coding.

### 4.1 LLMs as Agents Compared to Humans

This study examined whether LLMs can function as assistive qualitative coding agents and how their outputs compare to those generated by human coders. Across both test cases, the results demonstrate that GPT4o-mini is capable of producing codes that align closely with human-generated code (72-83%), particularly with respect to consistency, relevance, distinction and specificity. The quantitative CReDS evaluation shows strong alignment with human codes and even surpasses human codes in Test Case Two for certain dimensions. Human evaluation results echo this pattern; overall code match rates range from 75-83% across agents and test cases, with exact match percentages between 22-24% in Test Case One. In Test Case Two, the LLM's codes align with human codes 71.6% of inputs and 95.7% of LLM code lists are deemed appropriate to the text, indicating strong assistive potential despite occasional divergence. These findings are consistent with prior work suggesting that LLMs can approximate human indicative coding under constrained conditions (Dunivin, 2025; Raza et al., 2025; Montes et al., 2025).

However, several limitations in LLM performance emerged. Human evaluators identified that LLM-generated codes frequently missed aspects of human coding, with between 25-72% of input text missing sentiments captured by human coders. This pattern indicates that while LLMs are adept at identifying dominant or explicit themes, they are less reliable in capturing subtle themes, layered meanings, or affective nuances. This is further highlighted by the focused sarcasm analysis, which shows that the model's performance declines minimally on sarcastic inputs (-0.1% for LLM Relevance), with specificity scores lower for sarcastic versus non-sarcastic comments (-8.6%). These results suggest that GPT4o-mini can reliably generate consistent and relevant codes, yet its ability to capture narrowly defined, nuanced meanings, especially when irony is present, remains limited as compared to human coders. This aligns with Montes et al.'s conclusions, where they found that LLMs fragmented data unnecessarily, missed latent interpretations, and sometimes produced themes with unclear boundaries (2025).

This underscores the importance of human oversight in the coding process and highlights the LLM's ability as a complementary tool rather than a replacement for expert judgment. Furthermore, the results demonstrate the potential for iterative development and refinement of

LLM agents to progressively enhance coding accuracy. Prior work by Parfenova et al showed that humans outperform LLMs when labelling complex sentences, where LLMs perform comparatively better with simple quotations (2025). Continued methodological refinement and targeted training may therefore enable LLMs to more closely align with human coding performance on complex and nuanced textual inputs in the future.

## 4.2 Implementation, Performance and Implications of the CReDS Framework

One contribution of this study is the development and implementation of the CReDS framework as a structured approach to evaluating LLM codes at scale with minimal human ground truth. Using two test cases, this study demonstrated that the CReDS framework can capture meaningful differences in code quality across four dimensions: Consistency, Relevance, Distinction, and Specificity.

Results indicate that LLM-generated codes perform comparably to human-generated codes on Consistency, Relevance and Distinction, with near-identical median scores across both datasets. This suggests that GPT-4o-mini is capable of generating internally coherent code sets and avoiding excessive redundancy, even when producing multiple codes per input. Notably, extremely low relevance scores were rare: only 6 of 3225 (0.19%) LLM-generated codes received a zero relevance score. From an applied perspective, this suggests that CReDS is effective at identifying clearly problematic outputs while avoiding excessive penalisation of alternative but defensible interpretations.

Specificity emerged as the most challenging dimension for both the LLM and the framework itself. LLM-generated codes showed consistently lower mean specificity scores than human codes, and no LLM-generated codes achieved a perfect specificity score in Test Case Two. Importantly, this pattern appears to reflect limitations in LLM code generation, but also conservative scoring behaviour by the LLM acting as evaluator. The Relevance and Specificity evaluators consistently hesitated to assign extreme scores (zero or one), particularly perfect scores, whereas human evaluators were more willing to judge codes as sufficiently narrow and analytically useful. This cautious scoring behaviour suggests that the evaluative LLM may be underconfident in labelling outputs as optimal, even when human reviewers judge them as such.

While this conservative scoring limits direct numerical comparability between human and LLM scores, it does not undermine the utility of the framework. Rather, it highlights that CReDS functions effectively as a *diagnostic* tool, identifying relative strengths and weaknesses in generated codes even when absolute scores may be conservative. This aligns with the framework’s intended purpose as an evaluative scaffold rather than a definitive measure of qualitative quality.

Previous studies have confronted the concept of quality and validity in qualitative research, cautioning against using strict criteria for quality (Seale, 1999; Angen, 2000). The CReDS framework does seem to run counter to this caution as it introduced structured

dimensions for evaluating coding quality; however, it is not intended to impose criteria on qualitative analysis itself. Rather, CReDS is designed as a pragmatic evaluative scaffold for improving LLM-generated codes, particularly within iterative agent development workflows. Within these workflows, some degree of standardisation is necessary to diagnose model behaviour, compare outputs across iterations and guide refinement without replacing researcher judgements.

#### 4.3 Ethical Implementation

In recent years, LLMs have made significant breakthroughs and are consistently used by the public for everyday tasks. This accessibility has expanded their use cases far beyond their previous potential, allowing researchers to leverage the power of LLMs. This advancement has lowered the technical barrier required to utilise these models, but some researchers have concerns about using LLMs in qualitative research, including concerns about ethical use and removal of human oversight (Dunivin, 2024; Schroeder et al., 2025; Ornelas et al., 2025). Delegating aspects of interpretation to automated systems requires transparency about how prompts, model configurations and researcher decisions shape the resulting analysis. Future work should consider how to document these interpretive choices through reflexive memos or audit trails, ensuring accountability within hybrid human-AI methodologies. Equally, researchers must remain attentive to the potential for models to reproduce biases embedded in training data, particularly when analysing sensitive or identifying topics and data. Addressing these issues is essential to developing ethically sound and coherent practices for computationally assisted qualitative research.

#### 4.4 Positionality and Reflexivity

The research team participating in this project brings together diverse disciplinary backgrounds spanning computer science, public health and qualitative research. This multi-disciplinary composition has strongly influenced how we think about data, interpretation and coding practices. The author's training in quantitative and computational methods initially shaped the study's orientation towards efficiency, reproducibility and performance metrics. From this vantage point, LLMs appeared to offer a technical solution to the challenges of time-intensive qualitative analysis and an opportunity to accelerate coding without compromising accuracy.

At the outset, LLMs were considered as tools to automate part of the coding process, reflecting an epistemological stance rooted in measurement and optimisation. As the study progressed, our engagement with the data and the LLM models prompted a deeper reconsideration of what constitutes analysis. Through iterative experimentation, it became evident that, while LLMs could generate plausible and contextually relevant codes, their limitations were equally striking. The preliminary findings revealed that quantitative evaluations, such as cosine similarity, failed to capture the depth of the interpretive relationship visible to human coders.

This evolution also challenged our assumptions about what is defined as good qualitative coding. Initially, success was measured by the degree of alignment between LLM and human-generated codes, as if agreement equated to accuracy. Over time, it became clear that this framework penalised the very quality that makes qualitative inquiry valuable: the ability to notice and name new or divergent meanings. The realisation that additional, yet relevant topics could be counted against the model's performance forces a shift in thinking. Rather than viewing divergence as error, we began to see it as interpretive variation or a signal of potential conceptual insight. This recognition underscores a tension between computational efficiency and interpretive depth and raises critical questions about how qualitative rigour should be defined in hybrid human-AI contexts.

Our evolving perspective also reflected the interpretive plurality within our own team and beyond. Members grounded in public health tended to emphasise the practical benefits of LLMs for scaling and efficiency, while those with qualitative or sociological training stressed reflexivity, context and meaning making. Negotiating these perspectives became a methodological exercise in itself, shaping both the study's design and its conclusions. The process highlighted that integrating LLMs into qualitative research requires not only technical skill but also a willingness to interrogate the assumptions underlying both human and machine-generated interpretations.

Finally, as researchers work with generative models, we recognise our own positionality within a shifting methodological landscape. Our interactions with LLMs are not neutral but mediated by prompt design, interpretive expectations and disciplinary biases that influence what counts as good results. This study, therefore, represents not only empirical exploration but an act of reflexive inquiry; an attempt to understand how meaning is co-constructed between human analysts and machines.

#### 4.5 Strengths

This study offers several notable strengths. One of the key strengths is the use of multiple complementary analyses, providing flexibility to investigate unique and nuanced elements that may not have been captured in more constrained approaches. First, the identification of duplicated input text items that were coded by independent human coders, as well as each LLM agent, offers valuable insights into the consistency and replicability of LLM coders as compared to their human counterparts. Secondly, the exploratory analysis into the specificity of generated codes highlights one of the most pressing limitations of LLM-generated codes. This analysis was able to investigate what specific elements of specificity are present in high-scoring, or lacking in low-scoring, codes, offering insights for future refinement. Lastly, the exploratory analysis into the influence of sarcasm in the input text highlights the apparent success of GPT-4o-mini's ability to recognise and code appropriately when confronted with non-literal text, especially with prompt refinements.

Another important strength is the inclusion of human evaluation, allowing human reviewers to assess the generated codes as compared to the input text and to human-generated codes as ground truth. This enables the identification of both missing and additional meanings that were not captured by automated methods alone. This human verification process contributes to the growing literature on how computational and qualitative methods can be meaningfully combined without collapsing one into the other.

In summary, these strengths position this study as more than a technical exercise, representing an inquiry into how qualitative research itself can evolve in the context of computational innovation. The study's design, combining automated evaluation with human interpretation, illustrates how methodological pluralism can enhance both rigour and creativity. As such, LLMs serve as catalysts for rethinking traditional qualitative research practices.

#### 4.6 Limitations

Limitations of this study are closely tied to the limitations of LLMs themselves. Inconsistencies in results, hallucinations and limitations in meaning interpretation are inherent to LLMs, which may cause limitations on the reproducibility of this study and future application of LLMs in qualitative research (Deng et al., 2023). Limitations should also be taken into account in the automated evaluation framework presented in this study, where cosine similarity may fail to capture more nuanced semantic meanings. Cosine similarity, while commonly used to assess textual alignment, is a surface-level measure that does not fully capture the contextual and latent meanings (Steck et al., 2024). Steck, Ekanadham and Kallus emphasise that cosine similarity, though helpful for general alignment, can overlook nuanced thematic relationships, especially in qualitative contexts (2024). While the CReDS framework has deepened the evaluative process beyond just cosine similarity, some of the dimensions still rely on cosine similarity, particularly Distinction. Further work is still needed to robustly compare the similarity of generated codes.

This study also had limitations in methodology, specifically with a limited sample size due to the unstructured output of the LLM. This required data filtering and further reduction of the sample size, thereby reducing the quality of the output of the model described as well as introducing some small bias of what codes were eventually evaluated. While these effects were mitigated in the second test case, this is a consistent challenge to be considered in this work. Additionally, while GPT-4o mini represents an advanced LLM, it is still a closed, general-purpose system trained on a broad dataset rather than domain-specific materials (OpenAI, 2024). This limits transparency, interpretability and contextual responsiveness. Future studies might explore fine-tuned models that allow closer control of model behaviour and training data.

Finally, this study does not address researcher reflexivity when integrating computational systems into qualitative workflows. The introduction of LLMs inevitably reconfigures the relationship between human judgement, interpretation and methodological control. Addressing these tensions will be central to establishing ethical and methodologically sound practices for human-AI collaboration in qualitative research.

## 4.7 Impacts and Recommendations

This study contributes to the growing body of work exploring the integration of LLMs into qualitative research workflows. The findings suggest that models such as GPT-4o mini can generate meaningful initial codes that align well with conventional thematic analysis practices. However, this work highlights the importance of viewing LLMs not as replacements for human coders, but as tools to augment and support the traditional coding process (Khan et al, 2024). Used in conjunction with human expertise, LLMs can help streamline early stages of analysis by generating initial candidate codes and proposing alternative interpretations that may be overlooked by human coders. In doing so, they can help support saturation and broaden conceptual coverage, particularly in smaller teams or resource-limited settings. Furthermore, the CReDS framework can be positioned as a tool for iterative model development to align LLM-generated codes with qualitative research traditions and ensure code quality beyond alignment to the available human codebase.

Despite these strengths, the use of LLMs for qualitative analysis must be approached with caution. The interpretive depth and contextual sensitivity inherent to human-led coding remain essential, and LLM-generated outputs should be critically reviewed, refined and framed within the broader narrative of the research. LLM-generated outputs should therefore be treated as subject to critical scrutiny rather than as authoritative analytical facts. This reinforces the necessity of maintaining interpretive accountability, where the researcher remains actively engaged in examining assumptions that machine-generated codes are acting on. As the use of generative AI is normalised and introduced to qualitative analysis, this concept should remain critical.

Furthermore, evaluation of LLM-generated codes should move beyond surface-level quantitative metrics such as cosine similarity. While useful for benchmarking, these measures are limited in their ability to capture interpretive nuances. This limitation reflects the fundamentally qualitative nature of the field, where meaning, context and conceptual richness take precedence over statistical alignment. Additionally, this points to a broader methodological tension between computational reproducibility and qualitative depth. Evaluation frameworks may benefit from integrating quantitative elements, but they must remain grounded in the goals and epistemologies of qualitative research. Importantly, those interpreting the value of LLM-generated codes will ultimately prioritise the accuracy, specificity and analytical utility of the codes over their proximity to human-coded labels. While these may seem similar, this distinction is critical: alignment to human coders does not necessarily equate to high-quality coding. More important is whether the LLM meaningfully captures and represents relevant ideas within the context of the data and the research questions. This insight underscores the need for domain-sensitive, theory-informed evaluation approaches when incorporating LLMs into qualitative workflows. This summarises the motivations for the CReDS framework: utilising streamlined quantitative metrics that are grounded in the traditions of qualitative research.

## 4.8 Future Work

This study presents several promising directions for methodological and empirical development.

First, while the percentage of data points with at least one matching code achieves a maximal value of 83% alignment in the first test case, it does highlight the need to refine the generated codes. With further development, if the LLM were able to filter codes that were too general, the success would be much higher. A potential improvement could be the implementation of multi-agent peer review as suggested by Xu et al (2023). Collaboration between agents could reduce the number of “poor quality” codes by engaging multiple models to correct each other. An increased number of generated codes is not necessarily beneficial as previously thought; De Paoli found a similar result that generating more initial codes does not necessarily increase the accuracy of the analysis (2024). Further work should be done to continue to improve the LLM’s prompt to increase the quality of generated initial codes. Dunivin found significant improvements in model response by implementing a chain of thought prompting and further suggested that precise and specific descriptions within the prompt not only improved LLM coding but also improved the quality of human codes (2024). This acts as a clear support for implementing a human-LLM collaboration framework, as proposed by Dai, Xiong and Ku (2023).

Next, further work is needed to refine the metrics used in the CReDS framework. The results from this study show that these initial metrics are promising and seem to align with human evaluation. However, results from the exploratory analyses, particularly the specificity analysis, indicate that further calibration should be done for what a “perfect” score would look like. It is extremely rare for the CReDS framework to assign a perfect score to a code for any dimension, even when a human evaluator may rate it as an effective code. The rubrics provided to the LLM are potentially too strict, resulting in consistently lower scores. Further refinements and validation are still required.

By pursuing these directions, future research can move toward a more integrated understanding of how generative AI systems might participate in the co-construction of qualitative knowledge within traditional workflows, while retaining the critical reflexivity and contextual grounding that remain essential to qualitative methodological traditions.

## 5. Conclusions

This study examined the feasibility of using LLMs as assistive agents in qualitative coding and proposed a structured framework, CReDS, for evaluating the quality of LLM-generated codes without reliance on human ground truth. Situated within ongoing methodological debates about rigour, scalability and interpretive integrity in qualitative research, this work demonstrates that LLMs can meaningfully support early stages of thematic analysis when their outputs are carefully evaluated and situated within traditional human analytic workflows.

Using two real-world qualitative datasets, GPT4o-mini generated codes that aligned closely with those produced by human coders. Human evaluation showed that the majority of LLM-generated codes were appropriate to the input text, and a substantial proportion matched human meaning. At the text level, LLMs consistently identified at least one accurate code for most inputs, even when complete alignment with human coding was not achieved. These findings suggest that LLMs are effective at capturing dominant themes and analytically useful patterns, supporting their use as scalable tools for generating initial codes in large or resource-constrained qualitative projects. This study also highlights clear and persistent limitations in LLM-assisted coding. LLM-generated codes were less likely to capture subtle sentiment, narrow interpretive nuance and language features such as sarcasm. These limitations were reflected in both human evaluation and in lower specificity scores within the CReDS framework. These shortcomings

A key contribution of this thesis is the development and implementation of the CReDS framework as an alternative approach to evaluating qualitative codes. Unlike traditional intercoder agreement metrics or similarity-based comparisons, CReDS focuses on intrinsic code qualities, consistency, relevance, distinction and specificity, which align with long-standing qualitative methodological principles. Results demonstrate that CReDS can capture meaningful variation in code quality and identify strengths and weaknesses of LLM behaviour, even when absolute scoring is imperfect. While challenges remain in translating qualitative judgements into quantitative scores, this work shows that such an approach is both feasible and informative, particularly for interactive LLM development and diagnostic evaluation.

The findings also have broader implications for trustworthiness and transparency in LLM-assisted qualitative research. By making evaluative criteria explicit and operationalised, the CReDS framework supports a more accountable and reflexive use of AI tools. This approach encourages researchers to examine how codes are generated, scored and interpreted. In doing so, it aligns with qualitative commitments to reflexivity, interpretive rigour, and methodological transparency.

This study had several limitations. The evaluative LLM exhibited conservative scoring behaviour, particularly for Relevance and Specificity, suggesting that further refinement of evaluative prompts or refined training data may be necessary. The study also intentionally

avoided fine-tuning models on the evaluation data to prevent overfitting, positioning this work as an exploratory proof of concept rather than a finalised system. Despite these limitations, this thesis contributes to emerging conversations about human-AI collaboration in qualitative research. It demonstrates that LLMs can function as reliable assistive coding agents under defined conditions and that qualitative principles can inform the development of structured, scalable evaluation frameworks. Future work may build on these findings by refining CReDS metrics, integrating human and LLM scoring data for training purposes and extending evaluation to other qualitative methods and data types. Ultimately, this study supports a vision of qualitative analysis in which computational tools enhance, rather than diminish, the interpretive labour at the core of qualitative inquiry.

## References

- Ahmed, S. K., Mohammed, R. A., Nashwan, A. J., Ibrahim, R. H., Abdalla, A. Q., M. Ameen, B. M., & Khedhir, R. M. (2025). Using thematic analysis in qualitative research. *Journal of Medicine, Surgery, and Public Health*, 6, 100198.  
<https://doi.org/10.1016/j.glmedi.2025.100198>
- Angen, M. J. (2000). Evaluating Interpretive Inquiry: Reviewing the Validity Debate and Opening the Dialogue. *Qualitative Health Research*, 10(3), 378–395.  
<https://doi.org/10.1177/104973230001000308>
- Arthur, M. M. L. (2021). *Qualitative Coding*.  
<https://pressbooks.ric.edu/socialdataanalysis/chapter/qualitative-coding-2/>
- Babchuk, W. A. (2019). Fundamentals of qualitative analysis in family medicine. *Family Medicine and Community Health*, 7(2), e000040. <https://doi.org/10.1136/fmch-2018-000040>
- Barbour, R. S. (2001). Checklists for improving rigour in qualitative research: A case of the tail wagging the dog? *BMJ (Clinical Research Ed.)*, 322(7294), 1115–1117.  
<https://doi.org/10.1136/bmj.322.7294.1115>
- Baumgartner, J. M. (2025). *Pushshift/api* [Python]. <https://github.com/pushshift/api> (Original work published 2017)
- Bingham, A. J. (2023). From Data Management to Actionable Findings: A Five-Phase Process of Qualitative Data Analysis. *International Journal of Qualitative Methods*, 22, 16094069231183620. <https://doi.org/10.1177/16094069231183620>
- Boe, B. (n.d.). *PRAW 7.7.1 documentation*. Retrieved December 17, 2025, from <https://praw.readthedocs.io/en/stable/>
- Borchers, C., Shahrokhian, B., Balzan, F., Tajik, E., Sankaranarayanan, S., & Simon, S. (2025). *Temperature and Persona Shape LLM Agent Consensus With Minimal Accuracy Gains in Qualitative Coding* (arXiv:2507.11198). arXiv.  
<https://doi.org/10.48550/arXiv.2507.11198>
- Borse, N. S., Subramaniam, R. C., & Rebello, N. S. (2025). *Investigation of the Inter-Rater Reliability between Large Language Models and Human Raters in Qualitative Analysis* (arXiv:2508.14764). arXiv. <https://doi.org/10.48550/arXiv.2508.14764>
- Bradley, E. H., Curry, L. A., & Devers, K. J. (2007). Qualitative Data Analysis for Health Services Research: Developing Taxonomy, Themes, and Theory. *Health Services Research*, 42(4), 1758–1772. <https://doi.org/10.1111/j.1475-6773.2006.00684.x>
- Braun, V., & Clarke, V. (n.d.). *Using thematic analysis in psychology: Qualitative Research in Psychology: Vol 3, No 2*. Retrieved December 17, 2025, from <https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa>
- Campbell, J. L., Quincy, C., Osserman, J., & Pedersen, O. K. (2013). Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement. *Sociological Methods & Research*, 42(3), 294–320.  
<https://doi.org/10.1177/0049124113500475>
- Cena, E., Brooks, J., Day, W., Goodman, S., Rousaki, A., Ruby-Granger, V., & Seymour-Smith, S. (2024). Quality Criteria: General and Specific Guidelines for Qualitative Approaches in Psychology Research. A Concise Guide for Novice Researchers and Reviewers. *International Journal of Qualitative Methods*, 23, 16094069241282843.  
<https://doi.org/10.1177/16094069241282843>

- Charmaz, K. (2001). Qualitative Interviewing and Grounded Theory Analysis. In J. Gubrium & J. Holstein, *Handbook of Interview Research* (pp. 675–694). SAGE Publications, Inc. <https://doi.org/10.4135/9781412973588.n39>
- Chen, J., Lotsos, A., Cheng, S., Wang, C., Zhao, L., Hullman, J., Sherin, B., Wilensky, U., & Horn, M. (2025). *A Computational Method for Measuring “Open Codes” in Qualitative Analysis* (arXiv:2411.12142). arXiv. <https://doi.org/10.48550/arXiv.2411.12142>
- Coffey, A., & Atkinson, P. (1996). *Making sense of qualitative data: Complementary research strategies* (pp. x, 206). Sage Publications, Inc.
- Cosijn, E., & Ingwersen, P. (2000). Dimensions of relevance. *Information Processing & Management*, 36(4), 533–550. [https://doi.org/10.1016/S0306-4573\(99\)00072-2](https://doi.org/10.1016/S0306-4573(99)00072-2)
- Creswell, J. W. (2007). *Qualitative inquiry and research design: Choosing among five approaches, 2nd ed* (pp. xvii, 395). Sage Publications, Inc.
- Cruz, M., Patra, J., Fischer, B., Rehm, J., & Kalousek, K. (2007). Public opinion towards supervised injection facilities and heroin-assisted treatment in Ontario, Canada. *The International Journal on Drug Policy*, 18, 54–61. <https://doi.org/10.1016/j.drugpo.2006.12.001>
- Cui, Z., Li, N., & Zhou, H. (2025). A large-scale replication of scenario-based experiments in psychology and management using large language models. *Nature Computational Science*, 5(8), 627–634. <https://doi.org/10.1038/s43588-025-00840-7>
- Dai, S.-C., Xiong, A., & Ku, L.-W. (2023). *LLM-in-the-loop: Leveraging Large Language Model for Thematic Analysis* (arXiv:2310.15100). arXiv. <https://doi.org/10.48550/arXiv.2310.15100>
- Deng, J., Zubair, A., & Park, Y.-J. (2023). Limitations of large language models in medical applications. *Postgraduate Medical Journal*, 99(1178), 1298–1299. <https://doi.org/10.1093/postmj/qgad069>
- Dodgson, J. E. (2019). Reflexivity in Qualitative Research. *Journal of Human Lactation*, 35(2), 220–222. <https://doi.org/10.1177/0890334419830990>
- Drápal, J., Westermann, H., & Savelka, J. (2023). *Using Large Language Models to Support Thematic Analysis in Empirical Legal Studies* (arXiv:2310.18729). arXiv. <https://doi.org/10.48550/arXiv.2310.18729>
- Dunivin, Z. O. (2024). *Scalable Qualitative Coding with LLMs: Chain-of-Thought Reasoning Matches Human Performance in Some Hermeneutic Tasks* (arXiv:2401.15170). arXiv. <https://doi.org/10.48550/arXiv.2401.15170>
- Dunivin, Z. O. (2025). Scaling hermeneutics: A guide to qualitative coding with LLMs for reflexive content analysis. *EPJ Data Science*, 14(1), 28. <https://doi.org/10.1140/epjds/s13688-025-00548-8>
- Eakin, J. M., & Gladstone, B. (2020). “Value-adding” Analysis: Doing More With Qualitative Data. *International Journal of Qualitative Methods*, 19, 1609406920949333. <https://doi.org/10.1177/1609406920949333>
- Gadesha, V. (n.d.). *What is few shot prompting?* | IBM. Retrieved December 17, 2025, from <https://www.ibm.com/think/topics/few-shot-prompting>
- Gao, J., Shu, Z., Yeo, S. Y., Prakash, A., Huang, C.-M., Dredze, M., & Xiao, Z. (2025). *Efficiency with Rigor! A Trustworthy LLM-powered Workflow for Qualitative Data Analysis* (arXiv:2501.00775). arXiv. <https://doi.org/10.48550/arXiv.2501.00775>
- Gibson, W. (n.d.). *Working with Qualitative Data*. Retrieved December 18, 2025, from [https://www.academia.edu/1243232/Working\\_with\\_Qualitative\\_Data](https://www.academia.edu/1243232/Working_with_Qualitative_Data)

- Goffman, E. (2022). *Stigma: Notes on the management of spoiled identity*. Penguin Books.
- Halpin, S. N. (2024). Inter-Coder Agreement in Qualitative Coding: Considerations for its Use. *American Journal of Qualitative Research*, 8(3), 23–43. <https://doi.org/10.29333/ajqr/14887>
- Hila, A., & Hauser, E. (2025). *Assessing the Reliability of Large Language Models for Deductive Qualitative Coding: A Comparative Study of ChatGPT Interventions* (arXiv:2507.14384). arXiv. <https://doi.org/10.48550/arXiv.2507.14384>
- Hunt, G., Moloney, M., & Fazio, A. (2011). Embarking on large-scale qualitative research: Reaping the benefits of mixed methods in studying youth, clubs and drugs. *Nordisk Alkohol- & Narkotikatidskrift : NAT*, 28, 433–452. <https://doi.org/10.2478/v10199-011-0040-1>
- Jamieson, M., Govaart, G., & Pownall, M. (2023). Reflexivity in quantitative research: A rationale and beginner’s guide. *Social and Personality Psychology Compass*, 17. <https://doi.org/10.1111/spc3.12735>
- Katz, A., Fleming, G., & Main, J. (n.d.). *Thematic Analysis with Open-Source Generative AI and Machine Learning: A New Method for Inductive Qualitative Codebook Development*. Retrieved December 17, 2025, from <https://arxiv.org/html/2410.03721v1>
- Khan, A., Kegalle, H., D’Silva, R., Watt, N., Whelan-Shamy, D., & Magee, L. (n.d.). *Automating Thematic Analysis: How LLMs Analyse Controversial Topics*. Retrieved December 21, 2025, from <https://arxiv.org/html/2405.06919v1>
- Korstjens, I., & Moser, A. (n.d.). Series: Practical guidance to qualitative research. Part 2: Context, research questions and designs. *The European Journal of General Practice*, 23(1), 274–279. <https://doi.org/10.1080/13814788.2017.1375090>
- Lim, W. M. (2025). What Is Qualitative Research? An Overview and Guidelines. *Australasian Marketing Journal*, 33(2), 199–229. <https://doi.org/10.1177/14413582241264619>
- Link, B. G., & Phelan, J. C. (2001). Conceptualizing stigma. *Annual Review of Sociology*, 27, 363–385. <https://doi.org/10.1146/annurev.soc.27.1.363>
- Linneberg, M., & Korsgaard, S. (2019). Coding qualitative data: A synthesis guiding the novice. *Qualitative Research Journal*. <https://doi.org/10.1108/QRJ-12-2018-0012>
- Lumivero. (2025, September 4). Perfecting the art of coding qualitative data. Lumivero. <https://lumivero.com/resources/blog/perfecting-the-art-of-coding-qualitative-data/>
- MacCallum, N., & Lee, J. (n.d.). *GPT-4.1 Prompting Guide | OpenAI Cookbook*. Retrieved December 17, 2025, from [https://cookbook.openai.com/examples/gpt4-1\\_prompting\\_guide](https://cookbook.openai.com/examples/gpt4-1_prompting_guide)
- Macqueen, K., McLellan-Lemal, E., Kay, K., Milstein, B., & Cdc, A. (1998). Codebook Development for Team-Based Qualitative Analysis. *Cultural Anthropology*, 10, 31–36. <https://doi.org/10.1177/1525822X980100020301>
- Mathis, W., Zhao, S., Pratt, N., Weleff, J., & Paoli, S. D. (n.d.). *Inductive thematic analysis of healthcare qualitative interviews using open-source large language models: How does it compare to traditional methods? - PubMed*. Retrieved December 17, 2025, from <https://pubmed.ncbi.nlm.nih.gov/39067136/>
- McAlearney, A. S., Walker, D. M., Shiu-Yee, K., Crable, E. L., Auritt, V., Barkowski, L., Batty, E. J., Dasgupta, A., Goddard-Eckrich, D., Knudsen, H. K., McCrimmon, T., Scalise, A., Sieck, C., Wood, J., & Drainoni, M.-L. (2023). Embedding Big Qual and Team Science Into Qualitative Research: Lessons From a Large-Scale, Cross-Site Research Study.

- International Journal of Qualitative Methods*, 22, 16094069231165933.  
<https://doi.org/10.1177/16094069231165933>
- Mohammadi, M., Li, Y., Lo, J., & Yip, W. (2025). Evaluation and Benchmarking of LLM Agents: A Survey. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 6129–6139. <https://doi.org/10.1145/3711896.3736570>
- Montes, C. M., Feldt, R., Martos, C. M., Ouhbi, S., Premanandan, S., & Graziotin, D. (2025). *Large Language Models in Thematic Analysis: Prompt Engineering, Evaluation, and Guidelines for Qualitative Software Engineering Research* (arXiv:2510.18456). arXiv. <https://doi.org/10.48550/arXiv.2510.18456>
- Neuendorf, K. A. (2017). *The Content Analysis Guidebook*. SAGE Publications, Inc. <https://doi.org/10.4135/9781071802878>
- Nowell, L. S., Norris, J. M., White, D. E., & Moules, N. J. (2017). Thematic Analysis: Striving to Meet the Trustworthiness Criteria. *International Journal of Qualitative Methods*, 16(1), 1609406917733847. <https://doi.org/10.1177/1609406917733847>
- Nyaaba, M., Min, S., Abiswin Apam, M., Acheampong, K. O., Dwamena, E., & Zhai, X. (2025). *Optimizing Generative AI's Accuracy and Transparency in Inductive Thematic Analysis: A Human-AI Comparison* (SSRN Scholarly Paper 5174910). Social Science Research Network. <https://doi.org/10.2139/ssrn.5174910>
- O'Connor, C., & Joffe, H. (2020). Intercoder Reliability in Qualitative Research: Debates and Practical Guidelines. *International Journal of Qualitative Methods*, 19, 1609406919899220. <https://doi.org/10.1177/1609406919899220>
- Oksanen, J., Lucero, A., & Hämäläinen, P. (2025). *LLMCode: Evaluating and Enhancing Researcher-AI Alignment in Qualitative Analysis* (arXiv:2504.16671). arXiv. <https://doi.org/10.48550/arXiv.2504.16671>
- Olmos-Vega, F. M., Stalmeijer, R. E., Varpio, L., & Kahlke, R. (2023). A practical guide to reflexivity in qualitative research: AMEE Guide No. 149. *Medical Teacher*, 45(3), 241–251. <https://doi.org/10.1080/0142159X.2022.2057287>
- OpenAI. (n.d.). *GPT-4o mini: Advancing cost-efficient intelligence*. Retrieved December 17, 2025, from <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>
- Oranga, J., & Matere, A. (2023). Qualitative Research: Essence, Types and Advantages. *Open Access Library Journal*, 10(12), 1–9. <https://doi.org/10.4236/oalib.1111001>
- Ornelas, T., Araújo, A. A., Araújo, J., Araújo, M., Trinkenreich, B., & Kalinowski, M. (2025). *LLM-Assisted Thematic Analysis: Opportunities, Limitations, and Recommendations* (arXiv:2511.14528). arXiv. <https://doi.org/10.48550/arXiv.2511.14528>
- Paoli, S. D. (2024). Further Explorations on the Use of Large Language Models for Thematic Analysis. Open-Ended Prompts, Better Terminologies and Thematic Maps. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 25(3). <https://doi.org/10.17169/fqs-25.3.4196>
- Parfenova, A., Marfurt, A., Denzler, A., & Pfeffer, J. (2025). Text Annotation via Inductive Coding: Comparing Human Experts to LLMs in Qualitative Data Analysis. *Findings of the Association for Computational Linguistics: NAACL 2025*, 6456–6469. <https://doi.org/10.18653/v1/2025.findings-naacl.361>
- Parkington, K., Teferra, B. G., Rouleau-Tang, M., Perivolaris, A., Rueda, A., Dubrowski, A., Kapralos, B., Samavi, R., Greenshaw, A., Zhang, Y., Cao, B., Wu, Y., Rambhatla, S., Krishnan, S., & Bhat, V. (2025). *Human vs. LLM-Based Thematic Analysis for Digital*

- Mental Health Research: Proof-of-Concept Comparative Study* (arXiv:2507.08002). arXiv. <https://doi.org/10.48550/arXiv.2507.08002>
- Peeperkorn, M., Kouwenhoven, T., Brown, D., & Jordanous, A. (2024). *Is Temperature the Creativity Parameter of Large Language Models?* (arXiv:2405.00492). arXiv. <https://doi.org/10.48550/arXiv.2405.00492>
- Randerson, S., Graydon-Guy, T., Lin, E.-Y., & Casswell, S. (n.d.). *Exploring the Use of a Large Language Model for Inductive Content Analysis in a Discourse Network Analysis Study—Steve Randerson, Thomas Graydon-Guy, En-Yi Lin, Sally Casswell, 2025*. Retrieved December 17, 2025, from <https://journals.sagepub.com/doi/full/10.1177/08944393251326175>
- Raza, M. Z., Xu, J., Lim, T., Boddy, L., Mery, C. M., Well, A., & Ding, Y. (2025). *LLM-TA: An LLM-Enhanced Thematic Analysis Pipeline for Transcripts from Parents of Children with Congenital Heart Disease* (arXiv:2502.01620). arXiv. <https://doi.org/10.48550/arXiv.2502.01620>
- Reeping, D., Hampton, C., & Özkan, D. (2025). Interrogating the Use of Large Language Models in Qualitative Research Using the Qualifying Qualitative Research Quality Framework. *Studies in Engineering Education*, 6(2). <https://doi.org/10.21061/see.174>
- Roberts, J., Baker, M., & Andrew, J. (2024). *AI and Qualitative Research: The promise and perils of large language model (LLM) “assistance.”*
- Rotolo, B., Bin Noon, G., Chen, H. H., & Butt, Z. A. (2023). Public Attitudes Towards Vaccine Passports in Alberta During the “Pandemic of the Unvaccinated”: A Qualitative Analysis of Reddit Posts. *International Journal of Public Health*, 68, 1606514. <https://doi.org/10.3389/ijph.2023.1606514>
- Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2025). *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications* (arXiv:2402.07927). arXiv. <https://doi.org/10.48550/arXiv.2402.07927>
- Saldana, J. (n.d.). *Saldana, J. (2013). The Coding Manual for Qualitative Researchers (2nd ed.). London Sage. - References—Scientific Research Publishing*. Retrieved December 17, 2025, from <https://www.scirp.org/reference/referencespapers?referenceid=1752732>
- Schroeder, H., Aubin Le Quéré, M., Randazzo, C., Mimno, D., & Schoenebeck, S. (2025). Large Language Models in Qualitative Research: Uses, Tensions, and Intentions. *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–17. <https://doi.org/10.1145/3706598.3713120>
- Seale, C. (1999). Quality in Qualitative Research. *Qualitative Inquiry*, 5(4), 465–478. <https://doi.org/10.1177/107780049900500402>
- Seidel, J., & Kelle, U. (n.d.). *Seidel, J., & Kelle, U. (1995). Different functions of coding in the analysis of textual data. In U. Kelle (Ed.), Computer-dided qualitative data analysis Theory, methods, and practice. Thousand Oaks, CA Sage. - References—Scientific Research Publishing*. Retrieved December 18, 2025, from <https://www.scirp.org/reference/referencespapers?referenceid=168827>
- Sharma, A., Cochrane, K., & Wallace, J. R. (2025). *DeTAILS: Deep Thematic Analysis with Iterative LLM Support* (arXiv:2510.17575). arXiv. <https://doi.org/10.48550/arXiv.2510.17575>
- Shen, T. (2025). Grounded Theory Coding Quality Assurance Methods. *Methodology Insight*, 1, 1–15. <https://doi.org/10.71290/mi01010001>

- Steck, H., Ekanadham, C., & Kallus, N. (2024). Is Cosine-Similarity of Embeddings Really About Similarity? *Companion Proceedings of the ACM Web Conference 2024*, 887–890. <https://doi.org/10.1145/3589335.3651526>
- Stryker, C. (n.d.). *What Are Large Language Models (LLMs)?* | IBM. Retrieved December 17, 2025, from <https://www.ibm.com/think/topics/large-language-models>
- Stryker, C., & Holdsworth, J. (2024, August 11). *What Is NLP (Natural Language Processing)?* | IBM. <https://www.ibm.com/think/topics/natural-language-processing>
- Swert, K. D. (n.d.). *Calculating inter-coder reliability in media content analysis using Krippendorff's Alpha*.
- Timkey, W., & Schijndel, M. van. (2021). *All Bark and No Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality* (arXiv:2109.04404). arXiv. <https://doi.org/10.48550/arXiv.2109.04404>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. <https://doi.org/10.48550/arXiv.1706.03762>
- Wolski, U. (n.d.). “*The History of the Development and Propagation of QDA Software*” by Urszula Wolski PhD. Retrieved December 17, 2025, from <https://nsuworks.nova.edu/tqr/vol23/iss13/2/>
- Xu, Z., Shi, S., Hu, B., Yu, J., Li, D., Zhang, M., & Wu, Y. (2023). *Towards Reasoning in Large Language Models via Multi-Agent Peer Review Collaboration* (arXiv:2311.08152). arXiv. <https://doi.org/10.48550/arXiv.2311.08152>
- Yang, L. H., Kleinman, A., Link, B. G., Phelan, J. C., Lee, S., & Good, B. (2007). Culture and stigma: Adding moral experience to stigma theory. *Social Science & Medicine*, 64(7), 1524–1535. <https://doi.org/10.1016/j.socscimed.2006.11.013>
- Yi, S., Nguyen, J., Xu, H., Lim, T., Skrovan, J., Beri, M., Modi, H., Well, A., Leqi, L., Markey, M., & Ding, Y. (2025). *SFT-TA: Supervised Fine-Tuned Agents in Multi-Agent LLMs for Automated Inductive Thematic Analysis* (arXiv:2509.17167). arXiv. <https://doi.org/10.48550/arXiv.2509.17167>
- Zamawe, F. C. (2015). The Implication of Using NVivo Software in Qualitative Data Analysis: Evidence-Based Reflections. *Malawi Medical Journal: The Journal of Medical Association of Malawi*, 27(1), 13–15. <https://doi.org/10.4314/mmj.v27i1.4>
- Zhang, H., Wu, C., Xie, J., Rubino, F., Graver, S., Kim, C., Carroll, J. M., & Cai, J. (2024). *When Qualitative Research Meets Large Language Model: Exploring the Potential of QualiGPT as a Tool for Qualitative Coding* (arXiv:2407.14925). arXiv. <https://doi.org/10.48550/arXiv.2407.14925>
- Zhao, Z., & Liu, Y. (2025). *A Confidence-Diversity Framework for Calibrating AI Judgement in Accessible Qualitative Coding Tasks* (arXiv:2508.02029). arXiv. <https://doi.org/10.48550/arXiv.2508.02029>

## Appendices

### Appendix 1. Supplementary Tables and Figures

Figure i. Distribution of Original Test Case Two Dataset. This figure shows the original distribution of the word count for input texts for Test Case Two. This shows that the majority of data points had very short input texts, with few in the 100+ words category.

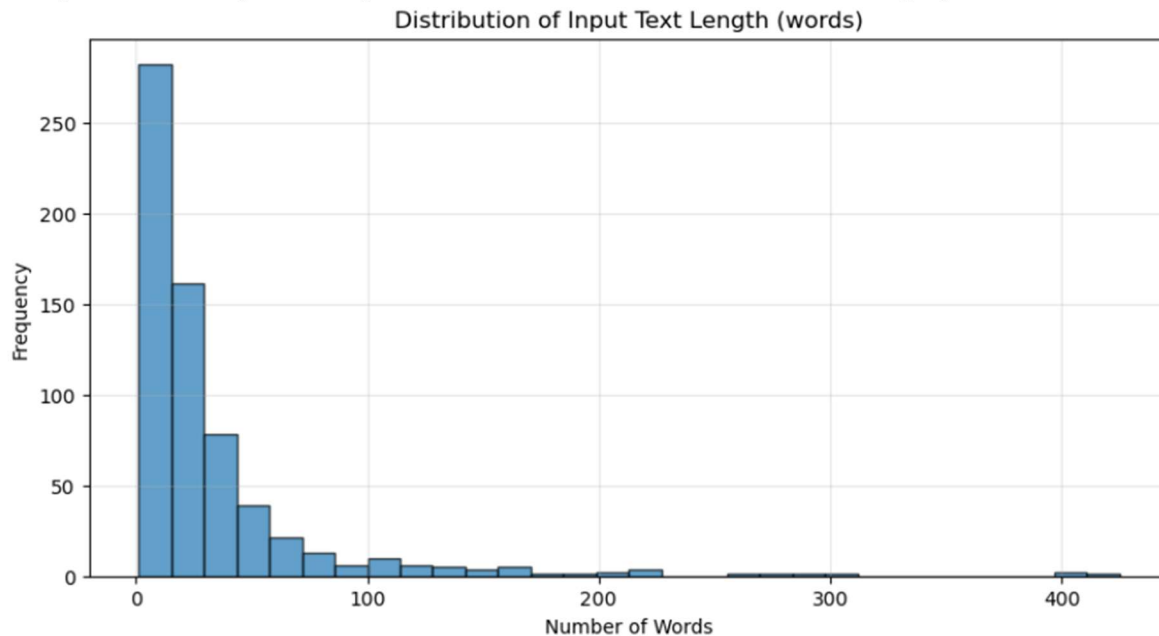


Figure ii. Distribution of Sampled Test Case Two Dataset. This figure shows the sampled distribution with respect to the word count for the input texts of Test Case Two. This shows a similar distribution to the original, where most of the sampled input text was very short, with few selected for 100+ word counts.

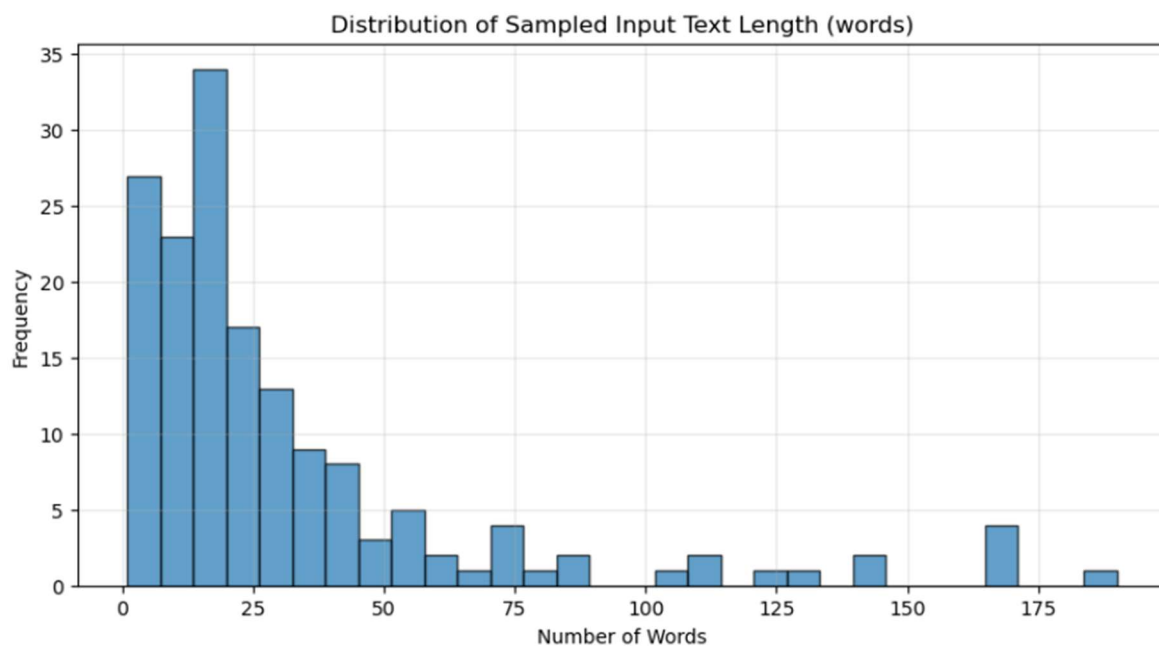


Figure iii. Distribution of Code-Original Code Similarity. This figure shows the distribution of the code to original code similarity for Test Case One across the three generated agents. This figure shows strong similarity between the selected agents, suggesting that temperature alone has little impact on the outcomes of code generation.

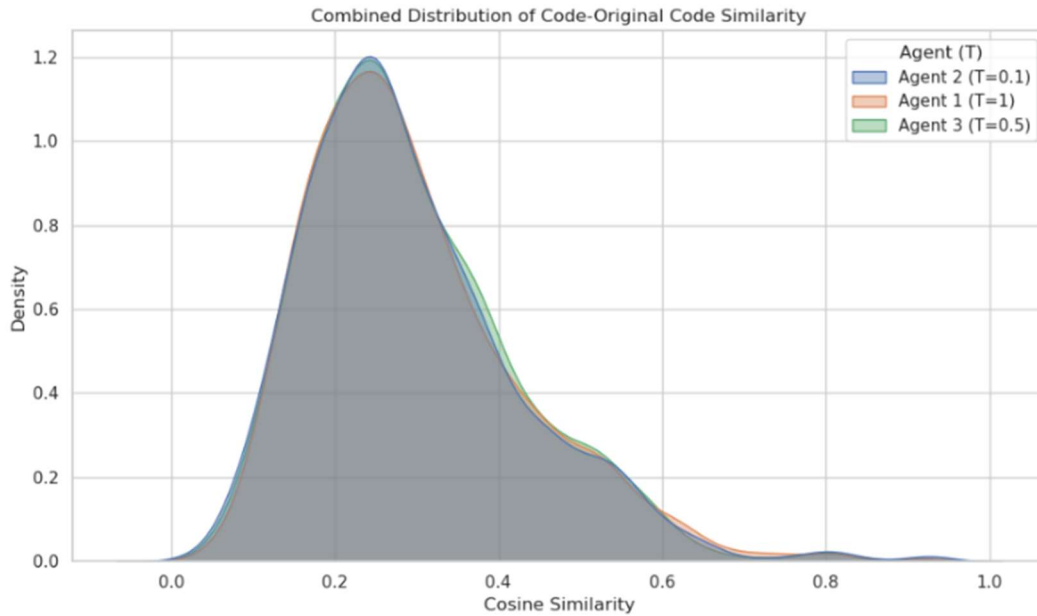


Figure iv. Distribution of Text- Explanation Similarity. This figure shows the distribution of the text to the explanation for Test Case One across the three generated agents. This figure again shows strong similarity between the selected agents, suggesting that temperature alone has little impact on the outcomes of code generation.

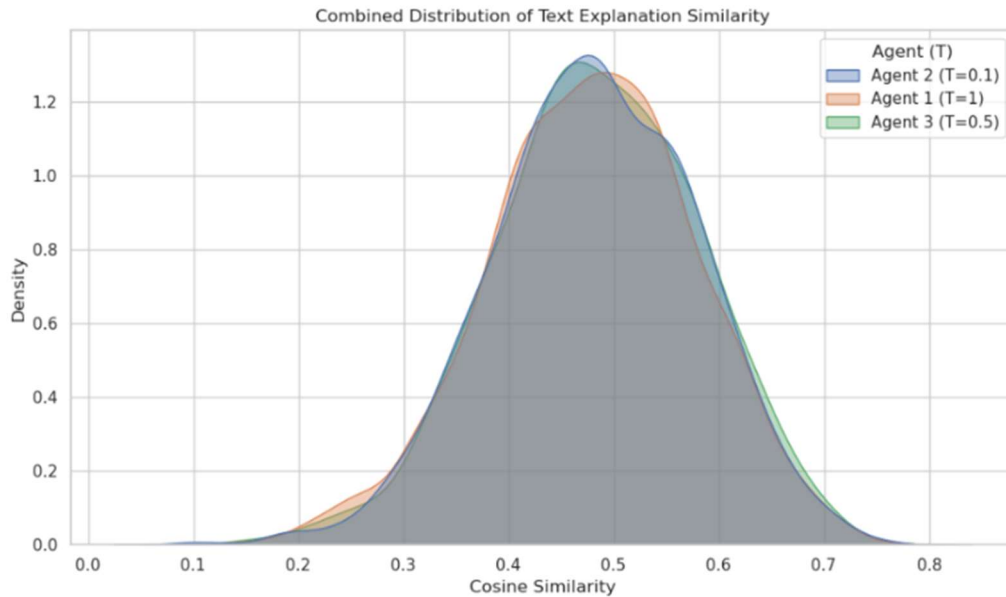


Figure v. Distribution of Text - Code Similarity. This figure shows the distribution of the code to original code similarity for Test Case One across the three generated agents. This figure shows strong similarity between the selected agents, suggesting that temperature alone has little impact on the outcomes of code generation.

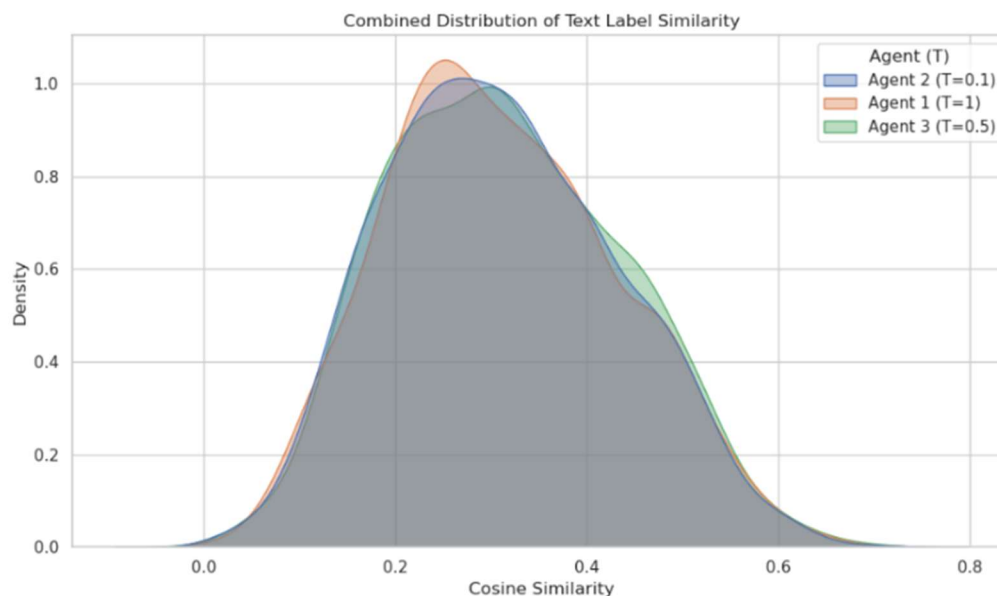


Figure vi. Boxplot of Code-Original Code Similarity by Agent. This figure, corresponding to Figure iii above, shows the distribution of values for the code to original code similarity. Again, this shows a nearly identical distribution between selected agents, suggesting that temperature has little or no impact on the similarity between LLM-generated codes and the original human codes.

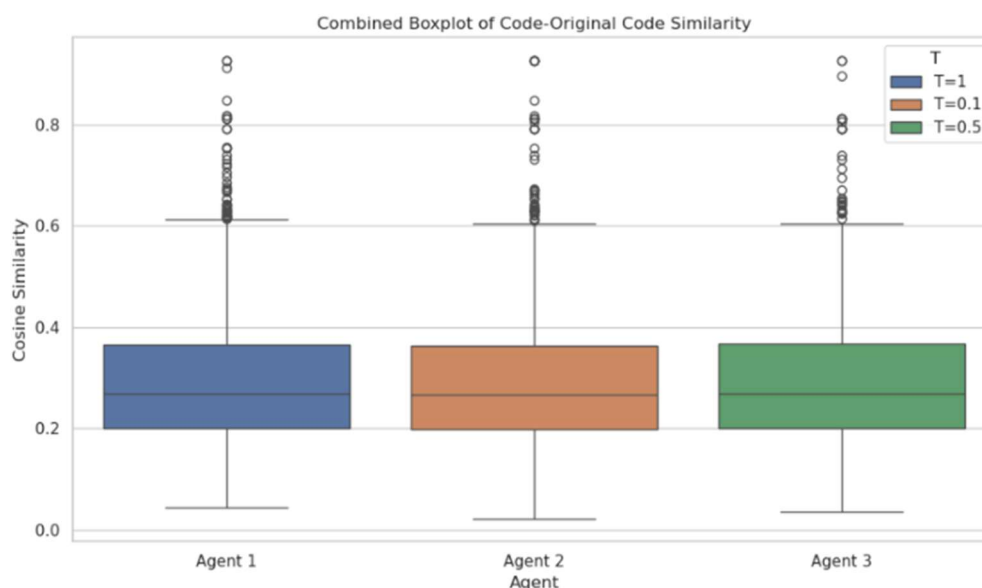


Figure vii. Boxplot of Input Text- Generated Explanation Similarity by Agent. This figure, corresponding to Figure iv above, shows the distribution of values for the text to explanation similarity. Again, this shows a nearly identical distribution between selected agents, suggesting that temperature has little or no impact on the similarity between LLM-generated codes and the original human codes.

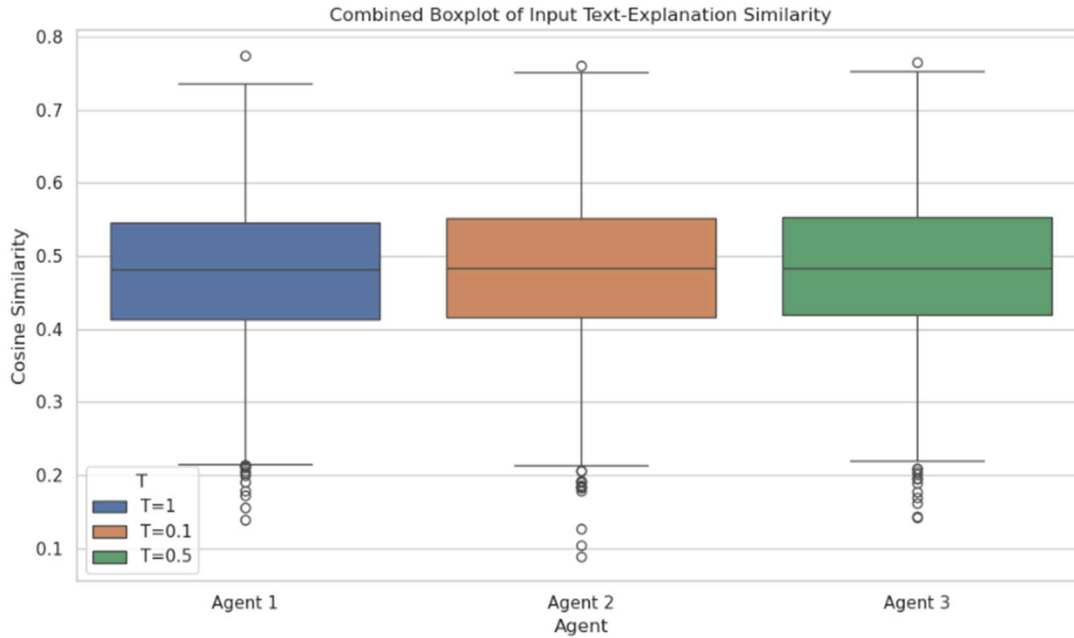
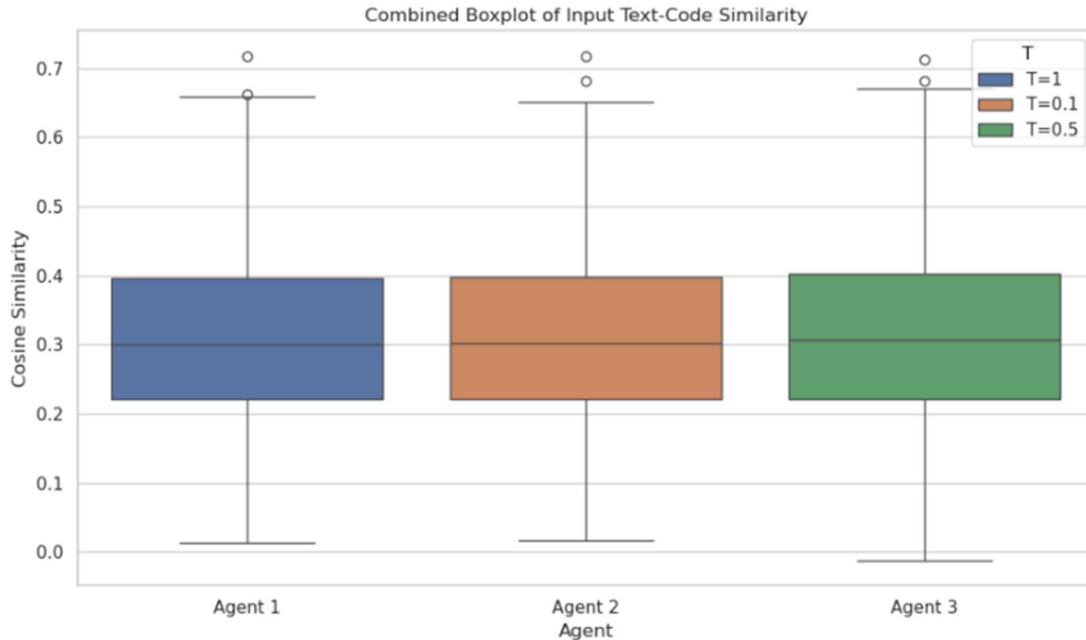


Figure viii. Boxplot of Input Text - Generated Code Similarity by Agent. Boxplot of Input Text-Code Similarity by Agent. This figure, corresponding to figure v above, shows the distribution of values for the text to code similarity. Again, this shows a nearly identical distribution between selected agents, suggesting that temperature has little or no impact on the similarity between LLM-generated codes and the original human codes.



## Appendix 2. Supplementary Background Information

### **Braun and Clarke framework:**

The Braun and Clarke framework includes six key phases of thematic analysis, which outline the common processes researchers follow. Braun and Clarke (2006) describe the following:

#### 1. Familiarising yourself with your data

Researchers are encouraged to immerse themselves in the data to an extent that they are familiar with the depth and breadth of the content. This includes actively reading, potentially multiple times, to search for meanings and patterns within the data. If working with verbal data, transcription into written form can be an excellent way to achieve familiarization with the data.

#### 2. Generating initial codes

This phase involves the generation of initial codes in order to begin organising data into meaningful groups. A code captures a salient feature of the data that appears relevant to researchers, but specific decision-making will hinge on whether the goal is data-driven or theory-driven. This framework also emphasises that coding should be done systematically across the entire dataset, either manually or using assistive computer software. Braun and Clarke highlight three key points of advice: 1) code for as many potential patterns as possible, 2) code extracts of data with consideration of the surrounding data's context and 3) individual extracts of data can be coded into as many themes as appropriate.

#### 3. Searching for themes

After all data has been initially coded, the third phase re-focuses the analysis at a broader level of themes. Analysing the codes, researchers should consider how the different codes may combine to form an overarching theme. Techniques include the use of tables, mindmaps or organisation using physical paper theme-piles. This phase ends with a collection of candidate themes and sub-themes to refine in the next phase.

#### 4. Reviewing themes

This phase involves two levels of reviewing and refining themes identified in the previous phase. Level one will review the themes at the level of the coded data extracts, ensuring that all collated extracts form a coherent pattern. Similarly, level two will review the themes but at the entire dataset level. This level considers the validity of the individual themes with respect to the dataset, in addition to whether the candidate thematic map reflects the meanings evident in the dataset. To some extent, the decision of accuracy and representativeness of themes is dependent on theoretical and analytical approaches specific to respective studies. The dataset is re-read to ascertain this validity, in addition to coding any additional data that were missed in earlier stages.

#### 5. Defining and naming themes

Assuming a satisfactory thematic map developed in stage four, the generated themes are defined and further refined to present for analysis. At this stage, researchers return to the collated data

extracts for each theme to analyse their content more deeply, aiming to articulate the central idea or "essence" of each theme. Rather than simply summarising or paraphrasing the data, researchers must identify what is significant about each theme in relation to the research question. This phase also includes considering how each theme relates to others and to the overall narrative of the analysis, ensuring clarity and minimising overlap. Where applicable, sub-themes may be identified to provide structure within complex themes and to reflect layers of meaning. By the end of this phase, each theme should be clearly defined and named, with concise labels that effectively capture its content and purpose.

## 6. Producing the report

Producing the Report is the final step, where the researcher brings together the analysis into a coherent, compelling narrative that answers the research question(s). This involves weaving together the themes to tell a story about the data in a way that is both analytical and interpretive. The goal is to clearly explain how the themes were identified, what they reveal and how they relate to existing literature or theoretical frameworks (if applicable).